How This Help File is Organized

VBA and Corel PHOTO-PAINT 10.0

Corel PHOTO-PAINT 10.0 VBA Help outlines the key components of the Visual Basic for Applications (VBA) programming environment as it relates to the Corel PHOTO-PAINT 10.0 application.

VBA is a subset of the Microsoft Visual Basic (VB) object-oriented programming environment. VBA uses the Visual Basic Editor interactive development environment and the VB programming language to enhance applications by manipulating the application's objects, exposed by its object model. VBA can access other applications from within Corel PHOTO-PAINT 10.0 by referencing that application's object model components.

Corel PHOTO-PAINT 10.0 VBA Help assumes that you have a basic knowledge of VBA and the VB programming environment. For more detailed instruction relating to these topics, please consult the "Microsoft Visual Basic for Applications Help" from the Help menu in the Visual Basic Editor.

Object Model Reference

Corel PHOTO-PAINT 10.0 VBA Help divides its object model references into several categories:

- Classes
- Enums
- Properties
- Methods
- Events
- Constants

Topics within each category are listed in alphabetical order for easy navigation and contain links to other categories as they relate to the current topic.

Sample Code

Corel PHOTO-PAINT 10.0 VBA Help provides sample code for all properties, methods, and events in the Corel PHOTO-PAINT 10.0 object model. Code samples are located in the main topic window for each property, method, and event.

The **Corel PHOTO-PAINT** application object is not referenced in the sample code. The application object does not need to be included when referencing objects within the application. If you are referencing the Corel PHOTO-PAINT 10.0 application with VBA from another application, the reference must include the Corel PHOTO-PAINT application object.

Legend

- Default property
- Read-Only property

Legend

- Default method

Introduction to Visual Basic for Applications

What is Visual Basic for Applications?

Visual Basic for Applications (VBA) is a subset of the Microsoft Visual Basic (VB) object-oriented programming environment. VBA uses the Visual Basic Editor interactive development environment and the VB programming language to enhance applications by manipulating the application's objects, exposed by its object model. VBA can access other applications from within Corel PHOTO-PAINT 10.0 by referencing that application's object model components.

VBA provides you with a set of tools that you can use to customize the graphical user interface of Corel applications, such as CorelDRAW and Corel PHOTO-PAINT. These tools allow you to process information and present data in an efficient and effect forum. Even though VBA uses the Visual Basic programming language, it is considered "for applications" because it is most often integrated into another application in order to customize the functionality of that application.

VBA allows you to record and run macros that automate a series of tasks within the Corel application.

What is the difference between Visual Basic, Visual Basic for Applications and VBScript?

The Microsoft Visual Basic programming system is an advanced set of programming tools that provides advanced functionality and components for the Microsoft Windows operating system and other windows-based programs. For example, with Visual Basic you can create application extensions (dll's) and stand-alone executable programs (exe's). You cannot create either of these components with VBA or VBScript.

VBA is also referred to as Visual Basic, Applications Edition. VBA is a subset of the Visual Basic programming language. It uses the programming structure of Visual Basic to manipulate objects of an object model, left exposed by an application like CorelDRAW or Corel PHOTO-PAINT. The manipulation of these objects results in small packets of code procedures within the application. These code procedures and resulting projects are called add ins.

VBScript is also referred to as Microsoft Visual Basic, Scripting Edition. VBScript is also a subset of the Visual Basic programming language. It is a web-based HTML document scripting language.

The Visual Basic Editor

Each document or image that you create with VBA has a corresponding Visual Basic for Applications project. For example, when you open a document in Corel PHOTO-PAINT, an accompanying Visual Basic project is opened up in the Visual Basic Editor. In order to customize your document with VBA coding procedures, you must open the project file in the Visual Basic Editor. To display the Editor, go to **Tools|Visual Basic|Visual Basic Editor** on the main menu in the application.

For more detailed information on constructing code procedures in the Visual Basic Editor, view the Help Topics located within the Visual Basic Editor.

VBA Programming Components

What is an Object Model?

The VBA programming language automates tasks within an application, by manipulating the components of the application's object model. An object model represents the hierarchy of objects within an application and their relationship to each other within the paradigm.

In CorelDRAW and Corel PHOTO-PAINT, the **Application** object represents the beginning of the object hierarchy. Starting with the Application object, you drill down and navigate through the object model until you find the desired object. To reference an object with Visual Basic code, you separate each level of the object hierarchy with the dot operator (.).

For example, "Corel PHOTO-PAINT.ActiveDocument" references the **Document** object, but separates the document class from the application class with the use of the dot operator. It is important to note that you do not need to include the application object when referencing objects in the object model of CorelDRAW and Corel PHOTO-PAINT.

What is the difference between a Class and an Object?

A class is a definition of an object, not a representation of the actual object. A class outlines the properties, methods, and events that apply to a type of object in CorelDRAW and Corel PHOTO-PAINT. It acts as a template for all objects of that type class. An object is an instance of a class.

For example, "Document" represents the **Document** class in Corel applications. However, "ActiveDocument" represents an object within that class because it makes specific reference to one object.

What is a Collection?

A collection is a group of objects that are similar in type. As objects, they share the same properties, methods, and events. They are uniquely identified within the collection by their index number or their name. Collection objects act in the same manner and are always plural.

For example, "Documents" represent the **Documents** collection class in Corel applications. However, "Documents.Item (1)" references the first document object in the collection.

What is a Property?

Each object within an object model is defined by a property, method, event, or a combination of each. A property is an adjective, and acts as an object attribute. It represents a characteristic quality of the object. Properties can be returned, set, or read-only. The provide information about the object.

For example, "ActiveDocument.Name" represents the **Name** property of a Document object.

What is a Method?

Each object within an object model is defined by a property, method, event, or a combination of each. A method is a verb, and is seen as the action of an object. It represents an act that can be performed by an object.

For example, "ActiveDocument.Close" represents the **Close** method of a Document object. It performs the act of closing the document in the application.

What is an Event?

Each object within an object model is defined by a property, method, event, or a combination of each. An event is a noun, and acts as something that takes place in an object. You write code for an object to respond to the act. Events are triggered by an action, such as a click, key press, or system timer.

For example, "ActiveDocument.AfterSave" event triggers an action in the **Document** object after it has been saved.

What is the difference between Enumerations and Constants?

Enumerations and Constants both represent fixed values in VBA programming structures. Unlike a variable, which temporarily stores a changing data value in a code procedure or function, enumerations and constants both represent fixed values in code procedures and functions - their values do not change.

An enumeration groups similar constants together. For Example, "AddinFilter" is an enumeration, yet it contains several constants, including "AddinFilterNone" and AddinFilterNew". A constant is an instance of an enumeration.

AddinHook properties

AddinHook Legend

- Application
  - Filter
  - Index
- Parent

AddinHook events

AddinHook

Execute
New

AddinHook

Class **AddinHook**

Properties Events Referenced By

{button ,AL(`CLS_AddinHook')} **Related Topics**

AddinHook.Application

Property **Application** As **Application**

Member of **AddinHook**

Read-Only

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddinHook;FNC_Application')} **Related Topics**

AddinHook.Filter

Property **Filter** As **pntAddinFilter**

Member of **AddinHook**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddinHook;FNC_Filter')} **Related Topics**

AddinHook.Index

Property **Index** As Long

Member of **AddinHook**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddinHook;FNC_Index')} **Related Topics**

AddinHook.Parent

Property **Parent** As **Application**

Member of **AddinHook**

Read-Only

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddinHook;FNC_Parent')} **Related Topics**

AddinHook.Execute

Event **Execute**()

Member of **AddinHook**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddinHook;FNC_Execute')} **Related Topics**

AddinHook.New

Event **New**(ByVal **NewDocument** As **Document**)

Member of **AddinHook**

Parameters	Description
NewDocument	Description of NewDocument goes here (**in**)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddinHook;FNC_New')} **Related Topics**

AddIns properties

AddIns Legend

- Application
- Collection
- Parent

AddIns methods

AddIns Legend

- Attach
- Detach
- DetachAll

AddIns

Class **AddIns**

Properties Methods Referenced By

{button ,AL(`CLS_AddIns')} **Related Topics**

AddIns.Application

Property **Application** As **Application**

Member of **AddIns**

Read-Only

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddIns;FNC_Application')} **Related Topics**

AddIns.Collection

Property **Collection** As Object

Member of **AddIns**

Read-Only

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddIns;FNC_Collection')} **Related Topics**

AddIns.Parent

Property **Parent** As **Application**

Member of **AddIns**

Read-Only

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddIns;FNC_Parent')} **Related Topics**

AddIns.Attach

Function **Attach**(ByVal **Filter** As **pntAddinFilter**, [ByVal ExecuteCommandPrompt As String]) As **AddinHook**

Member of **AddIns**

Parameters	Description
Filter	Description of Filter goes here (**in**)
ExecuteCommandPrompt	Description of ExecuteCommandPrompt goes here (**in**) Optional

Return value

Returns **AddinHook**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddIns;FNC_Attach')} **Related Topics**

AddIns.Detach

Sub **Detach**(ByVal **pAddinHook** As **AddinHook**)

Member of **AddIns**

Parameters	Description
pAddinHook	Description of pAddinHook goes here (**in**)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddIns;FNC_Detach')} **Related Topics**

AddIns.DetachAll

Sub **DetachAll**()

Member of **AddIns**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_AddIns;FNC_DetachAll')} **Related Topics**

Application properties

Application Legend

- ActiveDocument

- ActiveFrame
- ActiveLayer
- ActivePalette
- ActiveWindow
- ActiveWorkspace
- AddIns
- Application
- AppWindow
- Clipboard
- ConfigPath
- Documents
  - EventsEnabled
  - PaintColor
- Palettes
  - PaperColor
- Parent
- Path
- SetupPath
- VBE
- Version
- VersionBuild
- VersionMajor
- VersionMinor
  - Visible
- Windows
- Workspaces

Application methods

Application Legend

- CombineChannels
- CorelScript
- CorelScriptTools
- CreateBWColor
- CreateCMYColor
- CreateCMYKColor
- CreateColor
- CreateColorEx
- CreateDocument
- CreateFixedColor
- CreateGrayColor
- CreateHLSColor
- CreateHSBColor
- CreateLabColor
- CreateRegistrationColor
- CreateRGBColor
- CreateYIQColor
- OpenCorelScriptFile
- OpenDocument
- Quit

Application

Class **Application**

Properties Methods Referenced By

The **Application** class defines the characteristics of the main Corel PHOTO-PAINT application and describes the look and behavior of the application through its properties and methods.

The **Application** object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. With the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same **Application** object.

The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics. Corel PHOTO-PAINT puts the tools and supplies of a professional graphic design studio at your fingertips. With a click of the mouse, you can choose from a vast array of media and textures, unlimited colors, brushes of every shape and size, and a library of ready-made images. You can create your images from scratch, touch-up photographs, add text and special effects, and change the lighting that surrounds your subject. Corel PHOTO-PAINT provides hundreds of other fantastic features that you can use to imitate painting and photography techniques or to develop your own artistic style. You can also animate your images and share them with the world by publishing your work to the Internet.

Many elements in the work area give you control over the tools and features of Corel PHOTO-PAINT. Some of these elements include toolbars, the Property Bar, menu commands, and Docker windows. As you create your images, you can also access online Help topics at any time. If you require additional assistance, there are a variety of technical support services available.

{button ,AL(`CLS_Application')} **Related Topics**

Application.ActiveDocument

Property **ActiveDocument** As **Document**

Description

The **ActiveDocument** property returns a value associated with the current document in Corel PHOTO-PAINT. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

Although several documents may be open in Corel PHOTO-PAINT at one time, the active document is the document that is currently in use.

It is important to note that you do not need to activate a document before working with it in Corel PHOTO-PAINT. A document is automatically activated when it is referenced in a programming structure. For example:

```
Documents(1).Activate
```

{button ,AL(`CLS_Application;FNC_ActiveDocument')} **Related Topics**

Application.ActiveFrame

Property **ActiveFrame** As **Frame**

Description

The **ActiveFrame** property returns a value associated with the current frame in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

{button ,AL(`CLS_Application;FNC_ActiveFrame')} **Related Topics**

Application.ActiveLayer

Property **ActiveLayer** As **Layer**

Description

The **ActiveLayer** property returns a value associated with the active layer of Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

Once active, a layer is ready to receive any new objects you draw, import, or paste onto it.

The **ActiveLayer** property returns a Read-Only value.

{button ,AL(`CLS_Application;FNC_ActiveLayer')} **Related Topics**

Application.ActivePalette

Property **ActivePalette** As **Palette**

Description

The **ActivePalette** property returns a value associated with the current color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

You can use standard color collections like the Uniform Color Palette, customizable Color Palettes that you create and arrange, or color-matching systems like the PANTONE MATCHING SYSTEM.

It is important to note that even though Corel PHOTO-PAINT may have several color palettes, the active palette is the palette that is currently in use in Corel PHOTO-PAINT.

The **ActivePalette** property returns a Read-Only value.

Example

The following code example displays the name of the default palette, and the number of colors in that palette, in a message box:

```
Sub PaletteActive()
MsgBox "The default color palette is: " & ActivePalette.Name _
& "It contains " & ActivePalette.ColorCount & " colors."
End Sub
```

{button ,AL(`CLS_Application;FNC_ActivePalette')} **Related Topics**

Application.ActiveWindow

Property **ActiveWindow** As **Window**

Description

The **ActiveWindow** property returns a value associated with the active window in Corel PHOTO-PAINT. A window contains a Corel PHOTO-PAINT image. Corel PHOTO-PAINT may have several windows open at one time, but the active window is the window that is currently in use.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The **ActiveWindow** property returns a Read-Only value.

Example

The following code example refreshes the active window (updates it with the most recent information) in the current document:

```
Sub WindowActive()
ActiveWindow.Refresh
End Sub
```

{button ,AL(`CLS_Application;FNC_ActiveWindow')} **Related Topics**

Application.ActiveWorkspace

Property **ActiveWorkspace** As **Workspace**

Description

The **ActiveWorkspace** property returns a value associated with the current workspace in Corel PHOTO-PAINT. A workspace is a container for all the files that make up a project. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. Several workspaces may be available in the application, but the active workspace is the workspace that is currently in use.

You can create multiple workspaces for specific users or specific tasks and then apply them when required. You can also delete workspaces when they are no longer needed.

The **ActiveWorkspace** property returns a Read-Only value.

Example

The following code example displays the name of the active workspace in a message box:

```
Sub WorkspaceActive()
    MsgBox ActiveWorkspace.Name
End Sub
```

{button ,AL(`CLS_Application;FNC_ActiveWorkspace')} **Related Topics**

Application.AddIns

Property **AddIns** As **AddIns**

Gets Addins

Member of **Application**

Read-Only

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Application;FNC_AddIns')} **Related Topics**

Application.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. The application object refers to the Corel PHOTO-PAINT application. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, `CorelScript` and `Application.CorelScript` can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub ApplicationVersion()
    MsgBox "You are using: Corel PHOTO-PAINT " & Version
End Sub
```

{button ,AL(`CLS_Application;FNC_Application')} **Related Topics**

Application.AppWindow

Property **AppWindow** As **AppWindow**

Description

The **AppWindow** property returns a value associated with the application window of Corel PHOTO-PAINT. The application window is the area of Corel PHOTO-PAINT that contains the toolbars, workspaces, and windows. It is the container for the Corel PHOTO-PAINT application. Specifies attributes of the main application window in the Corel PHOTO-PAINT 10.0 application.

The **AppWindow** property returns a Read-Only value.

Example

The following code example displays the caption of the main application window in a message box:

```
Sub CaptionAppWindow()
MsgBox AppWindow.Caption
End Sub
```

{button ,AL(`CLS_Application;FNC_AppWindow')} **Related Topics**

Application.Clipboard

Property **Clipboard** As **Clipboard**

Description

The **Clipboard** property returns a value associated with the system clipboard on the local computer. The clipboard is a temporary storage area that is used to hold cut or copied information. The clipboard is always present in Windows and it stores information until it is replaced by another object or selection that has been cut or copied, or until you quit the current session of Windows.

The **Clipboard** property returns a Read-Only value.

Example

The following code example checks to see if there is any data in the system clipboard. If data is present, it is pasted into the current document. If no data is present, a message displays in a message box:

```
Sub ClipboardData()
If Not Clipboard.Empty Then   'if there is data in the clipboard
    ActiveLayer.Paste  'paste data into the active layer
Else
    MsgBox "There is no data in the clipboard."
End If
End Sub
```

{button ,AL(`CLS_Application;FNC_Clipboard')} **Related Topics**

Application.ConfigPath

Property **ConfigPath** As String

Description

The **ConfigPath** property returns a string value associated with the location of the configuration directory of Corel PHOTO-PAINT. The configuration directory contains the software settings used to customize an application. .

The **ConfigPath** property returns a Read-Only value.

Example

The following code example displays the path of the Corel PHOTO-PAINT configuration directory in a message box:

```
Sub PathConfig()
    MsgBox ConfigPath
End Sub
```

{button ,AL(`CLS_Application;FNC_ConfigPath')} **Related Topics**

Application.Documents

Property **Documents** As **Documents**

Description

The **Documents** property returns information about the **Documents** collection in Corel PHOTO-PAINT. The application object refers to the Corel PHOTO-PAINT application. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

The **Documents** collection defines the characteristics of document collection objects and describes the look and behavior of the collections objects through its properties, methods, and events.

A Document in Corel PHOTO-PAINT is a single page image, or a collection of single-page images that make up a multi-page document.

The **Documents** property returns a Read-Only value.

Example

The following code example counts the number of documents in the **Documents** collection:

```
Sub ApplicationDocument()
MsgBox Documents.Count
End Sub
```

{button ,AL(`CLS_Application;FNC_Documents')} **Related Topics**

A collection is a group of objects that have similar characteristics and actions but are uniquely identified by index names or numbers. Collections are always plural (i.e. Documents is a collection of document objects.)

Application.EventsEnabled

Property **EventsEnabled** As Boolean

Description

The **EventsEnabled** property returns a True or False value indicating the enabled status of an event in Corel PHOTO-PAINT. An event is enabled if it is ready and able to execute in the application. If an event is enabled, its **EventsEnabled** status is true. If the event is disabled, the status is set to false. Events are often disabled to temporarily improve performance and speed within Corel PHOTO-PAINT.

The **EventsEnabled** property returns a Boolean value.

Example

The following code example sets the **EventsEnabled** property to True:

```
Sub EnabledEvents()
Dim state As Boolean
state = EventsEnabled
End Sub
```

{button ,AL(`CLS_Application;FNC_EventsEnabled')} **Related Topics**

Application.PaintColor

Property **PaintColor** As **Color**

Description

The **PaintColor** property returns information about the paint color used in Corel PHOTO-PAINT. The paint color is the color used by the Paint tool to apply color and by the Shape Tools as an outline color.

{button ,AL(`CLS_Application;FNC_PaintColor')} **Related Topics**

Application.Palettes

Property **Palettes** As **Palettes**

Description

The **Palettes** property returns information about the **Palettes** collection in Corel PHOTO-PAINT. The **Palettes** collection defines the characteristics of Palette collection objects and describes the look and behavior of the collections objects through its properties, methods, and events.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

You can use standard color collections like the Uniform Color Palette, customizable Color Palettes that you create and arrange, or color matching systems like the PANTONE MATCHING SYSTEM.

The **Palettes** property returns a Read-Only value.

Example

The following code example displays the name of all palettes in use, and the number of colors in each palette, in a message box:

```
Sub PalettesList()
Dim pal As Palette
Dim s As String
s = "Active palettes: "
For Each pal In Palettes
    s = s & pal.Name & " (" & pal.ColorCount & " colors)"
Next pal
    MsgBox s
End Sub
```

{button ,AL(`CLS_Application;FNC_Palettes')} **Related Topics**

Application.PaperColor

Property **PaperColor** As **Color**

Gets or sets the paper color (used when clearing or erasing the image)

Member of **Application**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Application;FNC_PaperColor')} **Related Topics**

Application.Parent

Property **Parent** As **Application**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. The application object refers to the Corel PHOTO-PAINT application. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the application path in a message box:

```
Sub ParentApplication()
    MsgBox .Parent.Path
End Sub
```

{button ,AL(`CLS_Application;FNC_Parent')} **Related Topics**

Application.Path

Property **Path** As String

Description

The **Path** property returns a string value associated with the full path identity of the Corel PHOTO-PAINT directory. A path is the physical folder location of the application on a hard drive.

The **Path** property returns a Read-Only value.

Example

The following code example displays the path of the Corel PHOTO-PAINT directory in a message box:

```
Sub PathApplication()
MsgBox Path
End Sub
```

{button ,AL(`CLS_Application;FNC_Path')} **Related Topics**

Application.SetupPath

Property **SetupPath** As String

Description

The **SetupPath** property returns a string value associated with the location of the setup directory of Corel PHOTO-PAINT. The setup directory contains the software settings used to install an application. .

The **SetupPath** property returns a Read-Only value.

Example

The following code example displays the path of the Corel PHOTO-PAINT setup directory in a message box:

```
Sub PathSetup()
MsgBox SetupPath
End Sub
```

{button ,AL(`CLS_Application;FNC_SetupPath')} **Related Topics**

Application.VBE

Property **VBE** As Object

Description

The **VBE** property returns a reference to the VBA Editor in Corel PHOTO-PAINT. The VBA Editor is the component that creates, stores and edits VBA code modules in Corel PHOTO-PAINT. A VBA code module is a programming package of Visual Basic for Applications code segments.

VBA is a subset of the Microsoft Visual Basic (VB) object-oriented programming environment. VBA uses the Visual Basic Editor interactive development environment and the VB programming language to enhance applications by manipulating the application's objects, exposed by its object model. VBA can access other applications from within Corel PHOTO-PAINT 10.0 by referencing that application's object model components.

The **VBE** property returns a Read-Only value.

Example

The following code example displays the editor type name in the Debug window of the VBA Editor in Corel PHOTO-PAINT:

```
Sub AppVBE()
Dim obj as Object
Set obj = Application.VBE
    Debug.Print TypeName (obj)
End Sub
```

{button ,AL(`CLS_Application;FNC_VBE')} **Related Topics**

Application.Version

Property **Version** As String

Description

The **Version** property returns a value associated with the current version of Corel PHOTO-PAINT. A version is an edition of Corel PHOTO-PAINT - it is a particular form of the application, differing from previous editions. The **Version** property contains the version major number (before the decimal) and version minor number (after the decimal) of version numbers.

The **Version** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub AppVersion()
    MsgBox "You are using: " & Version
End Sub
```

{button ,AL(`CLS_Application;FNC_Version')} **Related Topics**

Application.VersionBuild

Property **VersionBuild** As Long

Description

The **VersionBuild** property returns a value associated with the current version of Corel PHOTO-PAINT. A version is an edition of Corel PHOTO-PAINT - it is a particular form of the application, differing from previous editions. The **VersionBuild** property returns only the build number of the current version. A build is specific edition of an application within a version in an application development environment.

The **VersionBuild** property is a Read-Only value.

Examples

The following code example displays the build number of the current Corel PHOTO-PAINT application in a message box:

```
Sub BuildVersion()
MsgBox "You are using Corel PHOTO-PAINT build " & VersionBuild
End Sub
```

{button ,AL(`CLS_Application;FNC_VersionBuild')} **Related Topics**

Application.VersionMajor

Property **VersionMajor** As Long

Description

The **VersionMajor** property returns a value associated with the current version of Corel PHOTO-PAINT. A version is an edition of Corel PHOTO-PAINT - it is a particular form of the application, differing from previous editions. The **VersionMajor** property returns only the first part of the version number (to the left of the decimal place).

For example, if the current version is 10.396, the major version value would be 10.

The **VersionMajor** property is a Read-Only value.

Example

The following code example checks the current version of Corel PHOTO-PAINT to ensure that the version number is at least 10.200:

```
Sub MajorVersion()
If VersionMajor = 10 And VersionMinor < 200 Then
    MsgBox "This script requires that Corel PHOTO-PAINT version 10.200 or later be installed."
End If
End Sub
```

{button ,AL(`CLS_Application;FNC_VersionMajor')} **Related Topics**

Application.VersionMinor

Property **VersionMinor** As Long

Description

The **VersionMinor** property returns a value associated with the current version of Corel PHOTO-PAINT. A version is an edition of Corel PHOTO-PAINT - it is a particular form of the application, differing from previous editions. The **VersionMinor** property returns only the second part of the version number (to the right of the decimal place).

For example, if the current version is 10.396, the minor version value is .396.

The **VersionMinor** property is a Read-Only value.

Example

The following code example checks the current version of Corel PHOTO-PAINT to ensure that the version number is at least 10.200:

```
Sub MinorVersion()
If VersionMajor = 10 And VersionMinor < 200 Then
    MsgBox "This script requires that Corel PHOTO-PAINT version 10.200 or later be installed."
End If
End Sub
```

{button ,AL(`CLS_Application;FNC_VersionMinor')} **Related Topics**

Application.Visible

Property **Visible** As Boolean

Description

The **Visible** property returns or sets a True or False value indicating the visibility status of Corel PHOTO-PAINT. The application object refers to the Corel PHOTO-PAINT application. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

If the **Visible** property is True, the application is running and visible on the monitor display. This property allows you to show or hide the Corel PHOTO-PAINT application window.

Example

The following code example hides Corel PHOTO-PAINT by setting the **Visible** property to False, and then shows Corel PHOTO-PAINT by setting the **Visible** property to True. Message boxes display messages indicating the visibility of the application:

```
Sub ApplicationVisible()
Visible = False
    MsgBox "Corel PHOTO-PAINT is hidden."
Visible = True
    MsgBox "Corel PHOTO-PAINT is now visible."
End Sub
```

{button ,AL(`CLS_Application;FNC_Visible')} **Related Topics**

Application.Windows

Property **Windows** As **Windows**

Description

The **Windows** property returns information about the **Windows** collection in Corel PHOTO-PAINT. The **Windows** collection defines the characteristics of windows collection objects and describes the look and behavior of the collections objects through its properties, methods, and events.

A Window contains a Corel PHOTO-PAINT image. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The **Windows** property returns a Read-Only value.

Examples

The following code example tiles all document windows horizontally:

```
Sub WindowsTile()
Windows.Arrange cdrTileHorizontally
End Sub
```

{button ,AL(`CLS_Application;FNC_Windows')} **Related Topics**

Application.Workspaces

Property **Workspaces** As **Workspaces**

Description

The **Workspaces** property returns information about the **Workspaces** collection in Corel PHOTO-PAINT. The **Workspaces** collection defines the characteristics of Workspace collection objects and describes the look and behavior of the collections objects through its properties, methods, and events.

A workspace is a container for all the files that make up a project. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. Several workspaces may be available in the application, but the active workspace is the workspace that is currently in use.

You can create multiple workspaces for specific users or specific tasks and then apply them when required. You can also delete workspaces when they are no longer needed.

The **Workspaces** property returns a Read-Only value.

Example

The following code example displays the available workspaces in a message box, with an asterisk next to the name of the active workspace:

```
Sub WorkspaceCollection()
Dim s As String
Dim wks As Workspace
s = "Available Workspaces: "
For Each wks In Workspaces
    s = s & wks.Name
    If wks.Active Then
    s = s & "  *"
    End If
Next wks
MsgBox s
End Sub
```

{button ,AL(`CLS_Application;FNC_Workspaces')} **Related Topics**

Application.CombineChannels

Function **CombineChannels**(ByVal **Mode** As **pntSplitMode**, ByVal **Channel1** As **Document**, ByVal **Channel2** As **Document**, ByVal **Channel3** As **Document**, [ByVal Channel4 As **Document** = 0], [ByVal CloseOriginals As Boolean = True], [ByVal Name As String]) As **Document**

Combine a set of channels into a new document

Member of **Application**

Parameters	Description
Mode	Description of Mode goes here (**in**)
Channel1	Description of Channel1 goes here (**in**)
Channel2	Description of Channel2 goes here (**in**)
Channel3	Description of Channel3 goes here (**in**)
Channel4	Description of Channel4 goes here (**in**) Optional Default value = **0**
CloseOriginals	Description of CloseOriginals goes here (**in**) Optional Default value = **True**
Name	Description of Name goes here (**in**) Optional

Return value

Returns **Document**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Application;FNC_CombineChannels')} **Related Topics**

Application.CorelScript

Function **CorelScript**() As Object

Description

The **CorelScript** method creates a Corel SCRIPT object. With Corel PHOTO-PAINT you can create, save, and run a script using the Corel SCRIPT language. This means that when working in Corel PHOTO-PAINT you can automate operations you use frequently by creating and saving a script. For more advanced script writing and editing, you can use the tools provided with the Corel SCRIPT Editor.

Although most Corel PHOTO-PAINT application commands are one-word equivalents of their corresponding menu commands, you might need more than the command itself to execute an action in these applications. If a command needs more information than is provided by the command name alone, parameters are required. Parameters represent aspects of the feature that you can change or selections you can make.

Corel SCRIPT programming statements and functions are a common set of instructions that can be used with any Corel application that supports scripting. Programming statements and functions are derived from traditional BASIC programming language dialects.

{button ,AL(`CLS_Application;FNC_CorelScript')} **Related Topics**

Application.CorelScriptTools

Function **CorelScriptTools**() As Object

Description

The **CorelScriptTools** method creates a CorelSCRIPT tool object. A script tool is implemented to assist with a particular Corel SCRIPT task. With Corel PHOTO-PAINT you can create, save, and run a script using the Corel SCRIPT language. This means that when working in Corel PHOTO-PAINT you can automate operations you use frequently by creating and saving a script. For more advanced script writing and editing, you can use the tools provided with the Corel SCRIPT Editor.

{button ,AL(`CLS_Application;FNC_CorelScriptTools')} **Related Topics**

Application.CreateBWColor

Function **CreateBWColor**(ByVal **White** As Boolean) As **Color**

Description

The **CreateBWColor** method creates a new color based on the Black and White color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Black and White color model is a 1-bit color mode that stores images as two solid colors - black and white - with no gradations. This color mode is useful for line art and simple graphics.

Parameters	Description
White	Sets the appearance of white in the Black and White color model. This value is a True or False value.

Example

```
CreateBWColor (True)
```

{button ,AL(`CLS_Application;FNC_CreateBWColor')} **Related Topics**

Application.CreateCMYColor

Function **CreateCMYColor**(ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long) As **Color**

Description

The **CreateCMYColor** method creates a new color based on the CMY color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMY is a color mode made up of cyan (C), magenta (M), and yellow (Y). This mode is used in the three-color printing process. In Corel applications, the CMY mode is the inverse of the RGB mode, with values ranging from 0 to 255. The CMY color mode is based on the CMY color model.

Parameters	Description
Cyan	Sets the level of cyan in the CMY color model. Values range from 0 to 255.
Magenta	Sets the level of magenta in the CMY color model. Values range from 0 to 255.
Yellow	Sets the level of magenta in the CMY color model. Values range from 0 to 255.

Example

```
CreateCMYColor (255, 100, 100)
```

{button ,AL(`CLS_Application;FNC_CreateCMYColor')} **Related Topics**

Application.CreateCMYKColor

Function **CreateCMYKColor**(ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long, ByVal **Black** As Long) As **Color**

Description

The **CreateCMYKColor** method creates a new color based on the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

Parameters	Description
Cyan	Sets the level of cyan in the CMY color model. Values range from 0 to 255.
Magenta	Sets the level of magenta in the CMY color model. Values range from 0 to 255.
Yellow	Sets the level of magenta in the CMY color model. Values range from 0 to 255.
Black	Sets the level of black in the CMYK color model. Values are percentages and range from 0 to 100.

Example

```
CreateCMYKColor (100, 0, 100, 0)
```

{button ,AL(`CLS_Application;FNC_CreateCMYKColor')} **Related Topics**

Application.CreateColor

Function **CreateColor**() As **Color**

Description

The **CreateColor** method creates a color object in Corel PHOTO-PAINT. When this method is used to create a color object, it creates the object with default color properties. You can change the default outline and fill colors by choosing a color when no object is selected. A dialog box prompts you to select the type of object for which you want to change the default color.

{button ,AL(`CLS_Application;FNC_CreateColor')} **Related Topics**

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The on-screen color palette is a toolbar that displays a series of color swatches. It is used to select colors for use in Corel PHOTO-PAINT. You can display multiple on-screen Color Palettes. They can be docked or left floating in the Application Window.

A parameter is synonymous with argument, a value that is passed to a routine. A parameter defines a characteristic of an object in the visual basic programming environment.

An index number is a reference to any collection that contains more than one object. It is used to identify each object in the collection. The index number can range from 1 to the number of available objects within the collection.

A color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

A channel is an 8-bit grayscale version of your image that functions like a plate used in the commercial printing process: each channel represents one level of color in your image. When all the channels are printed together, they produce the entire range of colors in the image.
For example, an RGB image comprises three channels (red, green, and blue). When all three channels are printed or displayed together, they create the entire range of colors in the image.

cdrPantone=1

cdrCMYK=2

cdrCMY=4

cdrRGB=5

cdrHSB=6

cdrHLS=7

cdrBlackAndWhite=8

cdrGray=9

cdrColorPantone=1

cdrColorCMYK=2

cdrColorCMY=4

cdrColorRGB=5

cdrColorHSB=6

cdrColorHLS=7

cdrColorBlackAndWhite=8

cdrColorGray=9

cdrColorYIQ=11

cdrColorLab=12

cdrColorPantoneHex=14

cdrColorRegistration=20

cdrColorSpot=25

cdrColorMixed=99

A macro is a collection of automatic tasks performing an action. It is a symbol, name or key that represents a list of commands.

A Document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A Document Page is sometimes called a "page." You can access a document page in a multi-page document by clicking on a page tab at the bottom-left corner of the Image Window.

Any script you create by saving a recording of your Corel PHOTO-PAINT operations is comprised of Corel SCRIPT application commands. Corel SCRIPT application commands instruct Corel PHOTO-PAINT to perform specific actions. For example, a command might instruct Corel PHOTO-PAINT to open or to close a document. The application commands are easy to understand, since most are one-word equivalents of the corresponding Corel application user interface.

Corel SCRIPT programming statements and functions send instructions or perform actions that are not part of another Corel application. For example, Corel SCRIPT programming statements can be used to display a custom dialog box, include flow control statements and constructs such as loops, create and manipulate variables, and retrieve information about your computer setup. On their own, Corel SCRIPT programming statements form a powerful programming language. A script containing only Corel SCRIPT programming statements can be executed even if another Corel application is not running.

A swatch is one of a series of solid-colored patches that is used as a sample when selecting color. A printed booklet of color swatches is called a swatch book. Swatch also refers to the colors contained in the Color Palette.

A data type consisting of a sequence of contiguous characters that represent the characters themselves rather than their numeric values. A String can include letters, numbers, spaces, and punctuation. The String data type can store fixed-length strings ranging in length from 0 to approximately 63K characters and dynamic strings ranging in length from 0 to approximately 2 billion characters. The dollar sign ($) type-declaration character represents a String in Visual Basic.

Application.CreateColorEx

Function **CreateColorEx**([ByVal Model As Long = 0], [ByVal c1 As Long = 0], [ByVal c2 As Long = 0], [ByVal c3 As Long = 0], [ByVal c4 As Long = 0], [ByVal c5 As Long = 0], [ByVal c6 As Long = 0], [ByVal c7 As Long = 0]) As **Color**

Description

The **CreateColorEx** method creates a new color, with specified color information, in Corel PHOTO-PAINT. This color is added to the active palette.

With the **CreateColorEx** method, you must specify the color model and components by passing them in a parameter:

Parameters	Description
ColorModel	The **ColorModel** parameter identifies the color model used to create the new color. This parameter specifies the numeric variable that is assigned to the color model. A color model is a simple color chart that defines the range of colors displayed in a color mode.
c1	The **c1** parameter specifies the numeric variable that is assigned to the first color component of the selected color model. For example, cyan is the first color component of the CMYK color model. The value is a percentage and can range from 1 to 100. A color that is selected at 100 is applied with full saturation.
c2	The **c2** parameter specifies the numeric variable that is assigned to the second color component of the selected color model. For example, magenta is the second color component of the CMYK color model. The value is a percentage and can range from 1 to 100. A color that is selected at 100 is applied with full saturation.
c3	The **c3** parameter specifies the numeric variable that is assigned to the third color component of the selected color model. For example, yellow is the third color component of the CMYK color model. The value is a percentage and can range from 1 to 100. A color that is selected at 100 is applied with full saturation.
c4	The **c4** parameter specifies the numeric variable that is assigned to the fourth color component of the selected color model. For example, black is the fourth color component of the CMYK color model. The value is a percentage and can range from 1 to 100. A color that is selected at 100 is applied with full saturation.
c5	The **c5** parameter specifies the numeric variable that is assigned to the fifth color component of the selected color model. Even though the most popular color models use only four levels of color saturations, Corel PHOTO-PAINT has the ability to create colors using seven color channels.
c6	The **c6** parameter specifies the numeric variable that is assigned to the sixth color component of the selected color model. Even though the most popular color models use only four levels of color saturations, Corel PHOTO-PAINT has the ability to create colors using seven color channels.
c7	The **c7** parameter specifies the numeric variable that is assigned to the seventh color component of the selected color model. Even though the most popular color models use only four levels of color saturations, Corel PHOTO-PAINT has the ability to create colors using seven color channels.

Example

The following code example adds a new color to the active color palette using the **CreateColorEx** method. The new color is added to the active color palette in Corel PHOTO-PAINT:

```
Sub CreatePaletteColor()
ActivePalette.AddColor CreateColorEx(2, 90, 90, 0, 0)
End Sub
```

{button ,AL(`CLS_Application;FNC_CreateColorEx')} **Related Topics**

Application.CreateDocument

Function **CreateDocument**(ByVal **Width** As Long, ByVal **Height** As Long, [ByVal Mode As **cdrImageMode** = cdrImageRGB (0)], [ByVal HRes As Long = 72], [ByVal VRes As Long = 72], [ByVal HasBackground As Boolean = True], [ByVal BackColor As **Color** = 0], [ByVal Movie As Boolean = False], [ByVal MovieFrames As Long = 1]) As **Document**

Creates a new document

Member of **Application**

Parameters	Description
Width	Description of Width goes here (**in**)
Height	Description of Height goes here (**in**)
Mode	Description of Mode goes here (**in**) Optional Default value = **cdrImageRGB** (0)
HRes	Description of HRes goes here (**in**) Optional Default value = **72**
VRes	Description of VRes goes here (**in**) Optional Default value = **72**
HasBackground	Description of HasBackground goes here (**in**) Optional Default value = **True**
BackColor	Description of BackColor goes here (**in**) Optional Default value = **0**
Movie	Description of Movie goes here (**in**) Optional Default value = **False**
MovieFrames	Description of MovieFrames goes here (**in**) Optional Default value = **1**

Return value

Returns **Document**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Application;FNC_CreateDocument')} **Related Topics**

Application.CreateFixedColor

Function **CreateFixedColor**(ByVal **PaletteID** As **cdrPaletteID**, ByVal **PaletteIndex** As Long, [ByVal Tint As Long = 100]) As **Color**

Description

The **CreateFixedColor** method creates a new color from a fixed color palette in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Parameters	Description
PaletteID	Sets the Palette ID that uniquely identifies which color palette to use in Corel PHOTO-PAINT.
PaletteIndex	Sets the color to use by referencing the index number of a color within a palette.)
Tint	Sets the tint of the palette color. Tints are lighter shades of a spot color that are created by changing the percentage tint value. This value is optional and the default value is 100.

Example

```
CreateFixedColor ( cdrPANTONECoated, 1, 50)
```

{button ,AL(`CLS_Application;FNC_CreateFixedColor')} **Related Topics**

Application.CreateGrayColor

Function **CreateGrayColor**(ByVal **GrayValue** As Long) As **Color**

Description

The **CreateGrayColor** method creates a new color from the Grayscale color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Grayscale color model is a color mode that displays images using 256 shades of gray. Each color is defined as a value between 0 and 255, where 0 is darkest (black) and 255 is lightest (white). In the RGB color mode, a grayscale value corresponds to equal amounts of all RGB colors; in CMYK, a grayscale value corresponds to zero C, M, and Y values, with a positive K value; in HSB, a grayscale value corresponds to zero H and S values, with a positive B value. The Grayscale color mode is based on the Grayscale color model.

Parameters	Description
GrayValue	Sets the level of gray in the Grayscale color model. Each color is defined as a value between 0 and 255, where 0 is darkest (black) and 255 is lightest (white).

Example

```
CreateGrayColor (255)
```

{button ,AL(`CLS_Application;FNC_CreateGrayColor')} **Related Topics**

Application.CreateHLSColor

Function **CreateHLSColor**(ByVal **Hue** As Long, ByVal **Lightness** As Long, ByVal **Saturation** As Long) As **Color**

Description

The **CreateHLSColor** method creates a new color from the HLS color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The HLS model is a variation of the HSB model and contains three components: hue, lightness, and saturation. Hue determines color (yellow, orange, red, etc.); lightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). The circular visual selector defines the H value (0 to 360) and the S value (0 to 100); the vertical visual selector defines the L value (0 to 100).

Parameters	Description
Hue	Sets the Hue for the HLS color model. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.
Lightness	Sets the Lightness for the HLS color model. Lightness determines the intensity of the color. Values range from 0 to 100.
Saturation	Sets the Saturation for the HLS color model. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.

Example

```
CreateHLSColor (200, 100, 100)
```

{button ,AL(`CLS_Application;FNC_CreateHLSColor')} **Related Topics**

Application.CreateHSBColor

Function **CreateHSBColor**(ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long) As **Color**

Description

The **CreateHSBColor** method creates a new color from the HSB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

HSB is a color model that approximates the way the human eye perceives color. In the HSB model, color is defined by three components: hue, saturation, and brightness. Hue determines color (yellow, orange, red, etc.); brightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). In the HSB color model, Hue (H) is expressed as a degree of rotation on a circular color wheel. Saturation (S) and brightness (B) are expressed as percentages of full intensity.

Parameters	Description
Hue	Sets the Hue for the HSB color model. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.
Saturation	Sets the Saturation for the HSB color model. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.
Brightness	Sets the Brightness for the HSB color model. Brightness is the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

Example

```
CreateHSBColor (200, 100, 100)
```

{button ,AL(`CLS_Application;FNC_CreateHSBColor')} **Related Topics**

Application.CreateLabColor

Function **CreateLabColor**(ByVal **L** As Long, ByVal **A** As Long, ByVal **B** As Long) As **Color**

Description

The **CreateLabColor** method creates a new color from the Lab color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The LAB color model is a color model created by the Commission Internationale de l'Eclairage (CIE). It contains a luminance (or lightness) component (L) and two chromatic components: "a" (green to red) and "b" (blue to yellow). The Lab color mode is based on the Lab color model.

Parameters	Description
L	Sets the Luminance, or brightness, value in the LAB color model. Values range from 0 to 255.
A	Sets one of two chromatic components in the LAB color model. "A" refers to a range of colors between green and red.
B	Sets one of two chromatic components in the LAB color model. "B" refers to a range of colors between blue and yellow.

Example

```
CreateLabColor (128, 0, 0)
```

{button ,AL(`CLS_Application;FNC_CreateLabColor')} **Related Topics**

Application.CreateRegistrationColor

Function **CreateRegistrationColor**() As **Color**

Description

The **CreateRegistrationColor** method creates a new color based on the Registration color model in Corel PHOTO-PAINT. Color trapping is necessary to compensate for poor color registration that occurs when the printing plates used to print each color, called color separations, are not aligned perfectly. Poor registration causes unintentional white slivers to appear between adjoining colors. Trapping is accomplished by intentionally overlapping colors so that minor problems with alignment are not noticed.

Example

```
CreateRegistationColor
```

{button ,AL(`CLS_Application;FNC_CreateRegistrationColor')} **Related Topics**

Application.CreateRGBColor

Function **CreateRGBColor**(ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long) As **Color**

Description

The **CreateRGBColor** method creates a new color from the RGB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

RGB is a color mode that contains three components: red (R), green (G), and blue (B). The RGB color mode is based on the RGB color model. In the RGB color mode, a value between 0 and 255 is assigned to each channel of red, green, and blue. An RGB color with the component values 0:25:118, for example, contains no red, some green, and more blue, resulting in a slightly greenish blue color. Monitors, scanners, and the human eye use RGB to produce or detect color.

Parameters	Description
Red	Sets the level of red in the RGB color model. Values range from 0 to 255.
Green	Sets the level of green in the RGB color model. Values range from 0 to 255.
Blue	Sets the level of blue in the RGB color model. Values range from 0 to 255.

Example

```
CreateRGBColor (255, 0, 0)
```

{button ,AL(`CLS_Application;FNC_CreateRGBColor')} **Related Topics**

Application.CreateYIQColor

Function **CreateYIQColor**(ByVal **Y** As Long, ByVal **I** As Long, ByVal **Q** As Long) As **Color**

Description

The **CreateYIQColor** method creates a new color from the YIQ color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The YIQ color model is a color model used in television broadcast systems (North American video standard - NTSC). Colors are split into a luminance value (Y) and two chromaticity values (I and Q). On a color monitor, all three components are visible. On a monochrome monitor, only the Y component is visible. The square, two-dimensional visual selector defines the I and Q values, and the vertical visual selector defines the Y value. All values are scaled from 0 to 255.

Parameters	Description
Y	Sets the luminance, or brightness, value in the YIQ color model. Values range from 0 to 255.
I	Sets one of two chromaticity values in the YIQ color model. In monitor calibration, chromaticity refers to the chroma (hue adjustment) of your monitor. Values range from 0 to 255.
Q	Sets one of two chromaticity values in the YIQ color model. In monitor calibration, chromaticity refers to the chroma (hue adjustment) of your monitor. Values range from 0 to 255.

Example

```
CreateYIQColor (100, 255, 255)
```

{button ,AL(`CLS_Application;FNC_CreateYIQColor')} **Related Topics**

Application.OpenCorelScriptFile

Function **OpenCorelScriptFile**(ByVal **FileName** As String) As **CorelScriptFile**

Description

The **OpenCorelScriptFile** method opens an existing Corel SCRIPT file and runs the script from a VBA macro. A macro is a collection of automatic tasks performing an action. It is a symbol, name or key that represents a list of commands.

With the **OpenCorelScriptFile** method, you must specify the script file by passing the file name in a parameter:

Parameters	Description
FileName	The **FileName** parameter identifies the full path name of the Corel SCRIPT file that is opened up in Corel PHOTO-PAINT and executed by converting it to a VBA macro.

{button ,AL(`CLS_Application;FNC_OpenCorelScriptFile')} **Related Topics**

Application.OpenDocument

Function **OpenDocument**(ByVal **FileName** As String, [ByVal Filter As **cdrFilter** = cdrAutoSense (0)], [ByVal LoadMode As **pntLoadMode** = pntLoadAll (0)], [ByVal Left As Long = 0], [ByVal Top As Long = 0], [ByVal Width As Long = 0], [ByVal Height As Long = 0], [ByVal DpiX As Long = 72], [ByVal DpiY As Long = 72], [ByVal StartFrame As Long = 1], [ByVal EndFrame As Long = 0]) As **Document**

Opens a document

Parameters	Description
FileName	Description of FileName goes here (**in**)
Filter	Description of Filter goes here (**in**) Optional Default value = **cdrAutoSense** (0)
LoadMode	Description of LoadMode goes here (**in**) Optional Default value = **pntLoadAll** (0)
Left	Description of Left goes here (**in**) Optional Default value = **0**
Top	Description of Top goes here (**in**) Optional Default value = **0**
Width	Description of Width goes here (**in**) Optional Default value = **0**
Height	Description of Height goes here (**in**) Optional Default value = **0**
DpiX	Description of DpiX goes here (**in**) Optional Default value = **72**
DpiY	Description of DpiY goes here (**in**) Optional Default value = **72**
StartFrame	Description of StartFrame goes here (**in**) Optional Default value = **1**
EndFrame	Description of EndFrame goes here (**in**) Optional Default value = **0**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Application;FNC_OpenDocument')} **Related Topics**

Application.Quit

Sub **Quit**()

Description

The **Quit** method closes the current session of Corel PHOTO-PAINT. You will be prompted to save any and all changes when you execute the **Quit** command. The application object refers to the Corel PHOTO-PAINT application. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a bitmap-based image-editing program that makes it easy to retouch existing photos or create original graphics.

Example

The following code example terminates the current session of Corel PHOTO-PAINT:

```
Sub QuitApp()
Quit
End Sub
```

{button ,AL(`CLS_Application;FNC_Quit')} **Related Topics**

AppWindow properties

AppWindow Legend

- Active

- Application
  - Caption

- ClientHeight

- ClientWidth

- Handle
  - Height
  - Left

- Parent
  - Top
  - Width
  - WindowState

AppWindow methods

AppWindow Legend

- Activate

AppWindow

Class **AppWindow**

Properties Methods Referenced By

The **AppWindow** class defines the characteristics of the application window object describes the look and behavior of the object through its properties and methods. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Application commands are accessible through the Menu Bar, toolbars, and flyouts. The Standard toolbar, located at the top of the Application Window by default, gives you access to some of the most common application commands, such as opening and saving images. The Property Bar and Docker windows provide access to commands that are relevant to the active tool or current task and can be opened, closed, and moved across your screen. You can also create multiple workspaces, which customize the look and feel of various Corel PHOTO-PAINT features. Workspaces are convenient if several people are using the same copy of Corel PHOTO-PAINT, or if you prefer to have different settings for different tasks.

There are several common features in the application window of Corel PHOTO-PAINT:

- Toolbox
- Property bar
- Workspaces
- Docker window

{button ,AL(`CLS_AppWindow')} **Related Topics**

cdrWindowNormal=1
cdrWindowMaximized=3
cdrWindowMinimized=2
cdrWindowRestore=9

AppWindow.Active

Property **Active** As Boolean

Description

The **Active** property returns a True or False value indicating the status of the Corel PHOTO-PAINT main application window. If the window is active, a value of true is returned, indicating that it is currently in use. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

If the window is inactive, the return value is false, indicating that the main application window is currently open but not in use.

Use the **Activate** method to change the inactive status of the main application window.

The **Active** property returns a Read-Only value.

Example

The following code example displays the status of the Corel PHOTO-PAINT main application window in a message box:

```
Sub AppWindowActive()
If AppWindow.Active Then
    MsgBox "Corel PHOTO-PAINT is currently active."
Else
    MsgBox "Corel PHOTO-PAINT is not active at this time."
End If
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Active')} **Related Topics**

AppWindow.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub VersionApplication()
With AppWindow
    MsgBox "You are using Corel PHOTO-PAINT " & Version
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Application')} **Related Topics**

AppWindow.Caption

Property **Caption** As String

Description

The **Caption** property returns or sets the display name of the Corel PHOTO-PAINT main application window. The caption appears as text in the title bar of the application window. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

If the document window is maximized when using the **Caption** property, the document title will also display after the caption in the title bar of the main application window.

Examples

The following code example sets the caption of the Corel PHOTO-PAINT main application window:

```
Sub AppWindowCaption()
AppWindow.Caption = "My PAINT Application"
End Sub
```

The following code example changes the existing caption of the Corel PHOTO-PAINT main application window:

```
Sub ResetCaption()
Dim NewCaption As String
NewCaption = "Corel PHOTO-PAINT"
If AppWindow.Caption <> NewCaption Then
    AppWindow.Caption = NewCaption
End If
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Caption')} **Related Topics**

AppWindow.ClientHeight

Property **ClientHeight** As Long

Description

The **ClientHeight** property returns a numerical value that measures the height of a client window within the main application window of Corel PHOTO-PAINT. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

The **ClientHeight** property returns a Read-Only value and is measured in screen pixels. A pixel is an abbreviation for picture element. Pixels are dots on a computer or television screen that combine to form an image. Computer images are created as an array of pixels, each having a specific color.

Example

The following example displays the client height, in screen pixels, in a message box:

```
Sub AppWindowClient()
With AppWindow
    MsgBox .ClientHeight
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_ClientHeight')} **Related Topics**

AppWindow.ClientWidth

Property **ClientWidth** As Long

Description

The **ClientWidth** property returns a numerical value that measures the width of a client window within the main application window of Corel PHOTO-PAINT. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

The **ClientWidth** property returns a Read-Only value and is measured in screen pixels. A pixel is an abbreviation for picture element. Pixels are dots on a computer or television screen that combine to form an image. Computer images are created as an array of pixels, each having a specific color.

Example

The following example displays the client width, in screen pixels, in a message box:

```
Sub AppWindowClient()
With AppWindow
    MsgBox .ClientWidth
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_ClientWidth')} **Related Topics**

AppWindow.Handle

Property **Handle** As Long

Description

The **Handle** property returns a value associated with the handle of the window in Corel PHOTO-PAINT. A handle allows you to resize the window in Corel PHOTO-PAINT.

The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application. This property returns a Read-Only value.

{button ,AL(`CLS_AppWindow;FNC_Handle')} **Related Topics**

AppWindow.Height

Property **Height** As Long

Description

The **Height** property sets the height of the Corel PHOTO-PAINT main application window on your desktop. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

This property only applies when the main application window is in a normal window state. This property will not work when the application window is minimized or maximized.

The height of the application window is a numerical value measured in screen pixels.

Example

The following code example sets the current window state to normal, defines the new dimensions of the window, resizes the Corel PHOTO-PAINT main application window according to the new dimensions:

```
Sub AppWindowHeight()
With AppWindow
    .WindowState = cdrWindowNormal
    .Top = 10
    .Left = 10
    .Height = 500
    .Width = 600
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Height')} **Related Topics**

AppWindow.Left

Property **Left** As Long

Description

The **Left** property sets the location of the left border of the Corel PHOTO-PAINT main application window on your desktop. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

This property only applies when the main application window is in a normal window state. This property will not work when the application window is minimized or maximized.

The **Left** property sets the distance, in screen pixels, from the Corel PHOTO-PAINT window's left border to the left side of the monitor's display area.

Example

The following code example sets the current window state to normal, defines the new dimensions of the window, resizes the Corel PHOTO-PAINT main application window according to the new dimensions:

```
Sub AppWindowLeft()
With AppWindow
    .WindowState = cdrWindowNormal
    .Top = 10
    .Left = 10
    .Height = 500
    .Width = 600
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Left')} **Related Topics**

AppWindow.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the path of the Corel PHOTO-PAINT application in a message box:

```
Sub AppWindowParent()
With AppWindow
    MsgBox .Parent.Application.Path
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Parent')} **Related Topics**

AppWindow.Top

Property **Top** As Long

Description

The **Top** property sets the location of the left border of the Corel PHOTO-PAINT main application window on your desktop. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

This property only applies when the main application window is in a normal window state. This property will not work when the application window is minimized or maximized.

The **Top** property sets the distance, in screen pixels, from the Corel PHOTO-PAINT window's top border to the top of the monitor's display area.

Example

The following code example sets the current window state to normal, defines the new dimensions of the window, resizes the Corel PHOTO-PAINT main application window according to the new dimensions:

```
Sub AppWindowTop()
With AppWindow
    .WindowState = cdrWindowNormal
    .Top = 10
    .Left = 10
    .Height = 500
    .Width = 600
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Top')} **Related Topics**

AppWindow.Width

Property **Width** As Long

Description

The **Width** property sets the height of the Corel PHOTO-PAINT main application window on your desktop. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

This property only applies when the main application window is in a normal window state. This property will not work when the application window is minimized or maximized.

The width of the application window is a numerical value measured in screen pixels.

Example

The following code example sets the current window state to normal, defines the new dimensions of the window, resizes the Corel PHOTO-PAINT main application window according to the new dimensions:

```
Sub AppWindowWidth()
With AppWindow
    .WindowState = cdrWindowNormal
    .Top = 10
    .Left = 10
    .Height = 500
    .Width = 600
End With
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Width')} **Related Topics**

AppWindow.WindowState

Property **WindowState** As **cdrWindowState**

Description

The **WindowState** property returns or sets the current window state of the Corel PHOTO-PAINT main application window. This window can be minimized, maximized or in a normal state. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

The **WindowState** property returns a value of cdr WindowState.

Example

The following code example begins with the application window maximized, sets the current window state to minimized, and displays a message that the window has been minimized. The main application window is then restored to a maximized state:

```
Sub AppWindowWindowState()
Dim state As cdrWindowState
state = cdrWindowMaximized    'window should be maximized to begin the procedure
AppWindow.WindowState = cdrWindowMinimized
MsgBox "Corel PHOTO-PAINT is now minimized."
AppWindow.WindowState = state
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_WindowState')} **Related Topics**

AppWindow.Activate

Sub **Activate**()

Description

The **Activate** method opens the main application window in Corel PHOTO-PAINT, if the window is not currently open. The application window represents the main Corel PHOTO-PAINT window - it acts as a container for the application.

In may cases, newer operating systems like Windows 98 and Windows 2000 don't allow the application to activate itself, to prevent intervention with you're current activity. If the application is not granted active status in certain situations, the application window will start flashing in the taskbar, but it will not appear in the foreground.

Example

The following code example activates the Corel PHOTO-PAINT main application window if it is currently inactive. A window is inactive if it is open, but it is not the currently used window application:

```
Sub Activate()
If Not AppWindow.Active Then
    AppWindow.Activate
End If
End Sub
```

{button ,AL(`CLS_AppWindow;FNC_Activate')} **Related Topics**

Background properties

Background Legend

- Active
- Application
- Parent

Background methods

Background Legend

- Activate
- Clear
- ConvertToLayer
- Copy
- Cut

Background

Class **Background**

Properties Methods Referenced By

The **Background** class defines the characteristics of the background objects and describes the look and behavior of the object through its properties and methods. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

You can control the way the colors of the objects and the image background are combined by choosing a merge mode. When you combine objects with the background, you can no longer change them as individual image elements. Combining objects with the image background merges them with the background and decreases file size.

When you create an object using the entire image background, you can move, arrange, and change the background. You can also create an object using part of the image background.

You can select a color mode and background color when you create a new image. If you want to add a background to an image without one, you can click Image, Create Background. The background is created using the paper color.

{button ,AL(`CLS_Background')} **Related Topics**

Background.Active

Property **Active** As Boolean

Description

The **Active** property returns a True or False value indicating the status of a background object in Corel PHOTO-PAINT. If a background is active, is it available for editing within the application. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

Use the **Activate** method to change the inactive status of the background object.

The **Active** property returns a Read-Only value.

{button ,AL(`CLS_Background;FNC_Active')} **Related Topics**

Background.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Background;FNC_Application')} **Related Topics**

Background.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Background;FNC_Parent')} **Related Topics**

Background.Activate

Sub **Activate**()

Description

The **Activate** method activates a background object in Corel PHOTO-PAINT. If a background is active, it is available for editing in the application. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

{button ,AL(`CLS_Background;FNC_Activate')} **Related Topics**

Background.Clear

Sub **Clear**([ByVal Color As **Color** = 0])

Description

The **Clear** method clears a background object in Corel PHOTO-PAINT. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

Parameters	Description
Color	Specifies a color to be removed from a background object. This value is optional and the default value is 0.

{button ,AL(`CLS_Background;FNC_Clear')} **Related Topics**

Background.ConvertToLayer

Function **ConvertToLayer**([ByVal Name As String], [ByVal Opacity As Long = 100], [ByVal MergeMode As **pntMergeMode** = pntMergeNormal (0)]) As **Layer**

Description

The **ConvertToLayer** method converts a background object into a layer in Corel PHOTO-PAINT. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Name	Sets the name of the converted layer. This value is a string value and it is optional.
Opacity	Sets the opacity level of the converted layer. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects. This value is optional and the default value is 100.
MergeMode	Sets the **MergeMode** of the converted layer. This value returns pntMergeMode. This value is optional and the default value is pntMergeNormal (0)

{button ,AL(`CLS_Background;FNC_ConvertToLayer')} **Related Topics**

Background.Copy

Sub **Copy**()

Description

The **Copy** method copies a background object in Corel PHOTO-PAINT. Copying the object places it in the system clipboard where is can be pasted at a later time. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

{button ,AL(`CLS_Background;FNC_Copy')} **Related Topics**

Background.Cut

Sub **Cut**([ByVal Color As **Color** = 0])

Description

The **Cut** method cuts a layer object in Corel PHOTO-PAINT. Cutting the object places it in the system clipboard where is can be pasted at a later time. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

Parameters	Description
Color	Specifies which color to cut from the background object with the Cut method. This value is optional and the default value is 0.

{button ,AL(`CLS_Background;FNC_Cut')} **Related Topics**

Channel properties

Channel Legend

- Active

- Application
  - Color
  - Index
  - InvertOverlay
  - Name
  - Opacity

- Parent
  - Visible

Channel methods

Channel Legend

- Activate
- CreateMask
- Delete
- StoreMask

Channel

Class **Channel**

Properties Methods Referenced By

The **Channel** class defines the characteristics of the channel objects and describes the look and behavior of the object through its properties and methods. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

A color channel is a type of channel that represents one component of an image's color model. Color channels are automatically generated by Corel PHOTO-PAINT when you create or open a color image file that has a 24-bit or 32-bit color depth. Individual channels include information about how much red, green, or blue is used in each image pixel, to produce the colors of the image. Combining all color channels displays the entire range of color present in the image.

A mask channel is a temporary storage area for masks. When you save a mask to an alpha channel, you can access and reuse it in the image as many times as you want. You can save an alpha channel to a file or load a previously saved channel in the active image.

Many of the features and commands that you can use to adjust the color and quality of an image are applied directly to the color channels that make up the image. Each image has one or more color channels that hold information about the color elements. The number of color channels in an image depends on the number of elements in the color model associated with the image. For example, an RGB image has three separate color channels, one for each color: red (R), green (G), and blue (B). The R, G, and B color channels store information about how much red, green, or blue is used in each pixel to produce the colors of the image.

When you view the combined color channels of an image, the resulting composite image displays the entire range of colors in the image. When you view color channels individually, you see a grayscale representation of the color information. A color channel can be edited and manipulated in the same way that you edit or manipulate a grayscale image.

Black-and-white, grayscale, duotone, and paletted images have only one color channel. RGB and Lab images have three channels, and CMYK images have four color channels.

{button ,AL(`CLS_Channel')} **Related Topics**

Channel.Active

Property **Active** As Boolean

Description

The **Active** property returns a True or False value indicating the status of a channel object in Corel PHOTO-PAINT. If a channel is active, is it available for editing within the application. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

Use the **Activate** method to change the inactive status of the channel object.

The **Active** property returns a Read-Only value.

{button ,AL(`CLS_Channel;FNC_Active')} **Related Topics**

Channel.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Channel;FNC_Application')} **Related Topics**

Channel.Color

Property **Color** As **Color**

Description

The **Color** property returns or sets the overlay color of a channel in Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

A color channel is a type of channel that represents one component of an image's color model. Color channels are automatically generated by Corel PHOTO-PAINT when you create or open a color image file that has a 24-bit or 32-bit color depth. Individual channels include information about how much red, green, or blue is used in each image pixel, to produce the colors of the image. Combining all color channels displays the entire range of color present in the image.

{button ,AL(`CLS_Channel;FNC_Color')} **Related Topics**

Channel.Index

Property **Index** As Long

Description

The **Index** property returns or sets an index value associated with a channel object in the **Channels** collection of Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

`Channels(5)` refers to the fifth channel object in the **Channels** collection of Corel PHOTO-PAINT.

{button ,AL(`CLS_Channel;FNC_Index')} **Related Topics**

Channel.InvertOverlay

Property **InvertOverlay** As Boolean

Description

The **InvertOverlay** property returns a True or False value that indicates whether or not a channel is inverted in the application. You can use the Invert filter to invert the color values of pixels in an image and create the appearance of a color photographic negative.

A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

{button ,AL(`CLS_Channel;FNC_InvertOverlay')} **Related Topics**

Channel.Name

Property **Name** As String

Description

The **Name** property returns or sets the name of a channel object in Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

{button ,AL(`CLS_Channel;FNC_Name')} **Related Topics**

Channel.Opacity

Property **Opacity** As Long

Description

The **Opacity** property returns the opacity value of a channel object's overlay in Corel PHOTO-PAINT. Opacity determines the intensity of a drop shadow. You can type values between 0 and 100. Low values create a less opaque overlay while high values create a more opaque overlay in the channel object.

A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

{button ,AL(`CLS_Channel;FNC_Opacity')} **Related Topics**

Channel.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Channel;FNC_Parent')} **Related Topics**

Channel.Visible

Property **Visible** As Boolean

Description

The **Visible** property returns a True or False value that indicates the visibility status of a channel object in Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

If the **Visible** property is set the True, the channel object is visible in Corel PHOTO-PAINT.

{button ,AL(`CLS_Channel;FNC_Visible')} **Related Topics**

Channel.Activate

Sub **Activate**()

Description

The **Activate** method activates a channel object in Corel PHOTO-PAINT. If a channel is active, it is available for editing in the application. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

{button ,AL(`CLS_Channel;FNC_Activate')} **Related Topics**

Channel.CreateMask

Sub **CreateMask**([ByVal MaskMode As **pntMaskMode** = pntMaskNormal (0)], [ByVal InvertMask As Boolean = False])

Description

The **CreateMask** method creates a new mask objects, based on the active channel object in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

Parameters	Description
MaskMode	Sets the **MergeMode** of the new mask. This value returns pntMergeMode. This value is optional and the default value is pntMergeNormal (0)
InvertMask	Sets a True or False value that indicates whether or not to invert the new mask. Often, it is easier to select the areas you want to protect from changes and then invert the selection. For example, if you want to edit a complex, irregularly shaped area, you can begin by selecting the surrounding area. Then, you can reverse the mask so that the area that was initially selected is protected, and the area that was masked is selected, i.e., available for editing. This value is optional and the default value is False.

{button ,AL(`CLS_Channel;FNC_CreateMask')} **Related Topics**

Channel.Delete

Sub **Delete**()

Description

The **Delete** method deletes a channel object in Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

{button ,AL(`CLS_Channel;FNC_Delete')} **Related Topics**

Channel.StoreMask

Sub **StoreMask**()

Description

The **StoreMask** method stores the current mask over a channel object in Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

{button ,AL(`CLS_Channel;FNC_StoreMask')} **Related Topics**

Channels properties

Channels Legend

- Application
- Count
-
- Item
- Parent

Channels methods

Channels Legend

- Add
- AddFromColorMask
- Load

Channels

Class **Channels**

Properties Methods Referenced By

The **Channels** class defines the characteristics of the **Channels** collection objects and describes the look and behavior of the object through its properties and methods. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

A color channel is a type of channel that represents one component of an image's color model. Color channels are automatically generated by Corel PHOTO-PAINT when you create or open a color image file that has a 24-bit or 32-bit color depth. Individual channels include information about how much red, green, or blue is used in each image pixel, to produce the colors of the image. Combining all color channels displays the entire range of color present in the image.

A mask channel is a temporary storage area for masks. When you save a mask to an alpha channel, you can access and reuse it in the image as many times as you want. You can save an alpha channel to a file or load a previously saved channel in the active image.

Many of the features and commands that you can use to adjust the color and quality of an image are applied directly to the color channels that make up the image. Each image has one or more color channels that hold information about the color elements. The number of color channels in an image depends on the number of elements in the color model associated with the image. For example, an RGB image has three separate color channels, one for each color: red (R), green (G), and blue (B). The R, G, and B color channels store information about how much red, green, or blue is used in each pixel to produce the colors of the image.

When you view the combined color channels of an image, the resulting composite image displays the entire range of colors in the image. When you view color channels individually, you see a grayscale representation of the color information. A color channel can be edited and manipulated in the same way that you edit or manipulate a grayscale image.

Black-and-white, grayscale, duotone, and paletted images have only one color channel. RGB and Lab images have three channels, and CMYK images have four color channels.

{button ,AL(`CLS_Channels')} **Related Topics**

Channels.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Channels;FNC_Application')} **Related Topics**

Channels.Count

Property **Count** As Long

Description

The **Count** property returns the number of channel objects in the **Channels** collection of Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

The **Count** property returns a Read-Only value.

{button ,AL(`CLS_Channels;FNC_Count')} **Related Topics**

Channels.Item

Property **Item**(ByVal **IndexOrName** As Variant) As **Channel**

Description

The **Item** property returns a value associated with the index number of a channel in the **Channels** collection of Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

`Channels(2)` refers to the second channel in the collection.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection.

Parameters	Description
IndexOrName	**Index** is a preset placeholder for each object in the Channels collection; it uniquely identifies each member of the collection. **Name** is the unique string value that identifies each channel.

{button ,AL(`CLS_Channels;FNC_Item')} **Related Topics**

Channels.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Channels;FNC_Parent')} **Related Topics**

Channels.Add

Function **Add**([ByVal Name As String], [ByVal Opacity As Long = 50], [ByVal Color As **Color** = 0], [ByVal InvertOverlay As Boolean = False], [ByVal FillWithBlack As Boolean = True]) As **Channel**

Description

The **Add** method adds a new channel to the Channels collection of Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

Parameters	Description
Name	Sets the name of the new channel. This value is a string value and it is optional.
Opacity	Sets the opacity level of the new channel. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects. This value is optional and the default value is 50.
Color	Sets the color of the new channel. This value is optional and the default value is 0.
InvertOverlay	Sets a True or False value that indicates whether or not to invert the overlay of the channel. This value is optional and the default value is False.
FillWithBlack	Sets a True or False value that indicates whether or not the new channel is filled with black. This value is optional and the default value is True.

{button ,AL(`CLS_Channels;FNC_Add')} **Related Topics**

Channels.AddFromColorMask

Function **AddFromColorMask**(ByVal **ColorMask** As **ColorMask**, [ByVal Name As String], [ByVal Opacity As Long = 50], [ByVal Color As **Color** = 0], [ByVal InvertOverlay As Boolean = False]) As **Channel**

Description

The **AddFromColorMask** method adds a new channel to the Channels collection, based on a color mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

Parameters	Description
ColorMask	Sets the color mask that is the basis for the new channel.
Name	Sets the name of the new channel. This value is a string value and it is optional.
Opacity	Sets the opacity level of the new channel. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects. This value is optional and the default value is 50.
Color	Sets the color of the new channel. This value is optional and the default value is 0.
InvertOverlay	Sets a True or False value that indicates whether or not to invert the overlay of the channel. This value is optional and the default value is False.

{button ,AL(`CLS_Channels;FNC_AddFromColorMask')} **Related Topics**

Channels.Load

Function **Load**(ByVal **FileName** As String, [ByVal Filter As **cdrFilter** = cdrAutoSense (0)], [ByVal LoadMode As **pntLoadMode** = pntLoadAll (0)], [ByVal Left As Long = 0], [ByVal Top As Long = 0], [ByVal Width As Long = 0], [ByVal Height As Long = 0]) As **Channel**

Description

The **Load** method loads a channel object from a file into Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

Parameters	Description
FileName	Specifies the computer location of the file containing the channel object that is loaded into Corel PHOTO-PAINT.
Filter	Specifies the filter to use when inserting the frame. This value returns cdrFilter. This value is optional and the default value is cdrAutoSense (0).
LoadMode	Sets the load mode of the channel. This value returns pntLoadMode. This value is optional and the default value is pntLoadAll (0).
Left	Sets the left position of the channel. This value is optional and the default value is 0.
Top	Sets the top position of the channel. This value is optional and the default value is 0.
Width	Sets the width of the channel. This value is optional and the default value is 0.
Height	Sets the height of the channel. This value is optional and the default value is 0.

{button ,AL(`CLS_Channels;FNC_Load')} **Related Topics**

Clipboard properties

Clipboard Legend

- Application
- Empty
- Parent
- Valid

Clipboard methods

Clipboard Legend

- Clear
- DataPresent

The clipboard is a temporary storage area that is used to hold cut or copied information. The clipboard is always present in Windows and it stores information until it is replaced by another object or selection that has been cut or copied, or until you quit the current session of Windows.

Clipboard

Class **Clipboard**

Properties Methods Referenced By

The **Clipboard** class defines the characteristics of Clipboard objects and describes the look and behavior of the objects through its properties and methods.

The clipboard is a temporary storage area that is used to hold cut or copied information. The clipboard is always present in Windows and it stores information until it is replaced by another object or selection that has been cut or copied, or until you quit the current session of Windows.

{button ,AL(`CLS_Clipboard')} **Related Topics**

Clipboard.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub VersionApplication()
    MsgBox "You are using Corel PHOTO-PAINT " & Version
End Sub
```

{button ,AL(`CLS_Clipboard;FNC_Application')} **Related Topics**

Clipboard.Empty

Property **Empty** As Boolean

Description

The **Empty** property returns a True or False value associated with presence of data in the system clipboard. Unlike the **Valid** property, which verifies the presence of a specific type or data (i.e. text, graphics), the **Empty** property looks for the presence of any data in the system clipboard.

The **Empty** property returns a Read-Only value.

Example

The following code example checks to see if there is any data in the clipboard. If there is data present, it is pasted in the active layer. If there is no data in the system clipboard, a message displays in a message box:

```
Sub ClipboardEmpty()
If Not Clipboard.Empty Then
   ActiveLayer.Paste
Else
   MsgBox "There is no data in the clipboard."
End If
End Sub
```

{button ,AL(`CLS_Clipboard;FNC_Empty')} **Related Topics**

Clipboard.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the caption of the clipboard's parent object, the main application window, in a message box:

```
Sub ClipboardParent()
With Clipboard
    MsgBox Parent.AppWindow.Caption
End With
End Sub
```

{button ,AL(`CLS_Clipboard;FNC_Parent')} **Related Topics**

Clipboard.Valid

Property **Valid** As Boolean

Description

The **Valid** property returns a True or False value associated with presence of valid data in the system clipboard. Valid information is anything that can be cut or copied into the clipboard, such as text and graphics selected within a document, or one or more files or folders.

The **Valid** property returns a Read-Only value.

Example

The following code example checks to see there is valid data in the clipboard. If there is valid data present, it is pasted into the active layer. If there is no valid data in the clipboard, a message displays in a message box:

```
Sub ClipboardValid()
If Clipboard.Valid Then
   ActiveLayer.Paste
Else
    MsgBox "There is no valid data in the clipboard."
End If
End Sub
```

{button ,AL(`CLS_Clipboard;FNC_Valid')} **Related Topics**

Clipboard.Clear

Sub **Clear**()

Description

The **Clear** method removes any and all data from the system clipboard. The clipboard should be cleared after large files, no longer needed, are pasted into a document. Clearing the clipboard of these large files frees up any system resources that the file is currently using in the clipboard.

Example

The following code example pastes the current contents of the system clipboard in the active document and uses the **Clear** method to remove the data from the clipboard:

```
Sub ClipboardClear()
ActiveLayer.Paste
Clipboard.Clear
End Sub
```

{button ,AL(`CLS_Clipboard;FNC_Clear')} **Related Topics**

Clipboard.DataPresent

Function **DataPresent**(ByVal **FormatName** As String) As Boolean

Description

The **DataPresent** method returns a True or False value that verifies the presence of a specific data format in the system clipboard. The data format (e.g. text, graphic, etc…) is determined by the Windows Clipboard Viewer utility. If you click the Display menu in the Viewer, all available data formats are available, with a check mark next to the active format. You may select other formats to change the current view of the data in the clipboard. The clipboard may contain more than one data format at a time.

Parameters	Description
FormatName	**FormatName** is a string parameter that determines the presence of a particular data format in the clipboard. It must be contained in quotes when passed to the function. "Text" is an example of a string parameter that may be passed as a **FormatName** parameter.

Example

The following code example verifies the presence of text data in the clipboard. It text data is present, the clipboard is cleared of any data with the Clear method. If there is no text data in the clipboard, a message displays in a message box:

```
Sub PresentData()
If Clipboard.DataPresent("Text") Then
   Clipboard.Clear
Else
    MsgBox "There is no text data in the clipboard."
End If
End Sub
```

{button ,AL(`CLS_Clipboard;FNC_DataPresent')} **Related Topics**

ClipMask properties

ClipMask Legend

- Application
  - Linked
- Parent
  - Visible

ClipMask methods

ClipMask Legend

- Apply
- Delete

ClipMask

Class **ClipMask**

Properties Methods Referenced By

The **ClipMask** class defines the characteristics of clip mask objects and describes the look and behavior of the object through its properties and methods. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A Clip mask lets you change the transparency of pixels in an object without permanently affecting the pixels of an object. You can create a clip mask to change the transparency of an entire object or to change the transparency of parts of an object. A clip mask also lets you change the transparency of clipping groups the same way that you change the transparency of an object. You can disable the clip mask of an object to temporarily hide the transparency of the object, or you can remove a clip mask to permanently remove the transparency of an object.

{button ,AL(`CLS_ClipMask')} **Related Topics**

ClipMask.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_ClipMask;FNC_Application')} **Related Topics**

ClipMask.Linked

Property **Linked** As Boolean

Description

The **Linked** property returns or sets a True or False value that indicates whether or not a clip mask object is linked in Corel PHOTO-PAINT. A clip mask is linked to an object when a plus sign is displayed between the thumbnails of the clip mask and the object in the Objects Docker window.

A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

{button ,AL(`CLS_ClipMask;FNC_Linked')} **Related Topics**

ClipMask.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_ClipMask;FNC_Parent')} **Related Topics**

ClipMask.Visible

Property **Visible** As Boolean

Description

The **Visible** property returns a True or False value that indicates the visibility status of a clip mask object in Corel PHOTO-PAINT. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

If the **Visible** property is set the True, the clip mask object is visible in Corel PHOTO-PAINT.

{button ,AL(`CLS_ClipMask;FNC_Visible')} **Related Topics**

ClipMask.Apply

Sub **Apply**()

Description

The **Apply** method applies a clip mask to an object in Corel PHOTO-PAINT. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

{button ,AL(`CLS_ClipMask;FNC_Apply')} **Related Topics**

ClipMask.Delete

Sub **Delete**()

Description

The **Delete** method deletes a clip mask object in Corel PHOTO-PAINT. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

{button ,AL(`CLS_ClipMask;FNC_Delete')} **Related Topics**

Color properties

Color Legend

- Application
  - BW
  - CMYCyan
  - CMYKBlack
  - CMYKCyan
  - CMYKMagenta
  - CMYKYellow
  - CMYMagenta
  - CMYYellow
  - Gray
  - HLSHue
  - HLSLightness
  - HLSSaturation
  - HSBBrightness
  - HSBHue
  - HSBSaturation
  - LabComponentA
  - LabComponentB
  - LabLuminance
- Name
- PaletteID
  - PaletteIndex
- Parent
  - RGBBlue
  - RGBGreen
  - RGBRed
  - Tint
- Type
  - YIQChromaI
  - YIQChromaQ
  - YIQLuminanceY

Color methods

Color Legend

- BWAssign
- CMYAssign
- CMYKAssign
- ConvertToBW
- ConvertToCMY
- ConvertToCMYK
- ConvertToFixed
- ConvertToGray
- ConvertToHLS
- ConvertToHSB
- ConvertToLab
- ConvertToRGB
- ConvertToYIQ
- CopyAssign
- CorelScriptAssign
- CorelScriptGetComponent
- FixedAssign
- GrayAssign
- HLSAssign
- HSBAssign
- LabAssign
- RegistrationAssign
- RGBAssign
- UserAssign
- UserAssignEx
- YIQAssign

Color

Class **Color**

Properties Methods Referenced By

The **Color** class defines the characteristics of color objects and describes the look and behavior of the collection objects through its properties and methods.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

There are many different color models that define colors - HSB, RGB, CMYK, and CIE Lab color models. The RGB and CMYK color models are only two of a number of models developed to suit a variety of digital design and desktop publishing applications. It is not necessary to be familiar with all these models, but it is helpful to be familiar with a few of the more widely used ones.

We all see color differently. Color is subjective to the human eye. Each device that interacts with your project's file: the scanner, monitor, and printer may have a different color space. For example, a color that is visible to the human eye may not be reproducible by your printer.

Because there are so many color variations, a precise method for defining each color is required. For example, once you find the perfect shade of light orange, you need to be able to reproduce that color and possibly tell others how to do the same. A color model defines that perfect shade of light orange by breaking it down into precise components that allow you to accurately transmit the information to other people and to the electronic devices you use to create projects.

{button ,AL(`CLS_Color')} **Related Topics**

Color.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub ColorsApp()
With ActivePalette.Color (1)
    MsgBox "You are using Corel PHOTO-PAINT " & .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Color;FNC_Application')} **Related Topics**

Color.BW

Property **BW** As Boolean

Description

The **BW** property returns or sets a True or False value that indicates the black and white state in the Black and White color model of Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Black and White color model is a 1-bit color mode that stores images as two solid colors - black and white - with no gradations. This color mode is useful for line art and simple graphics.

{button ,AL(`CLS_Color;FNC_BW')} **Related Topics**

Color.CMYCyan

Property **CMYCyan** As Long

Description

The **CMYCyan** property returns or sets the cyan color in the CMY color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMY is a color mode made up of cyan (C), magenta (M), and yellow (Y). This mode is used in the three-color printing process. In Corel applications, the CMY mode is the inverse of the RGB mode, with values ranging from 0 to 255. The CMY color mode is based on the CMY color model.

{button ,AL(`CLS_Color;FNC_CMYCyan')} **Related Topics**

Color.CMYKBlack

Property **CMYKBlack** As Long

Description

The **CMYKBlack** property returns or sets the back color value in the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

{button ,AL(`CLS_Color;FNC_CMYKBlack')} **Related Topics**

Color.CMYKCyan

Property **CMYKCyan** As Long

Description

The **CMYKCyan** property returns or sets the cyan color value in the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

{button ,AL(`CLS_Color;FNC_CMYKCyan')} **Related Topics**

Color.CMYKMagenta

Property **CMYKMagenta** As Long

Description

The **CMYKYellow** property returns or sets the yellow color value in the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

{button ,AL(`CLS_Color;FNC_CMYKMagenta')} **Related Topics**

Color.CMYKYellow

Property **CMYKYellow** As Long

Description

The **CMYKYellow** property returns or sets the yellow color value in the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

{button ,AL(`CLS_Color;FNC_CMYKYellow')} **Related Topics**

Color.CMYMagenta

Property **CMYMagenta** As Long

Description

The **CMYMagenta** property returns or sets the magenta color in the CMY color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMY is a color mode made up of cyan (C), magenta (M), and yellow (Y). This mode is used in the three-color printing process. In Corel applications, the CMY mode is the inverse of the RGB mode, with values ranging from 0 to 255. The CMY color mode is based on the CMY color model.

{button ,AL(`CLS_Color;FNC_CMYMagenta')} **Related Topics**

Color.CMYYellow

Property **CMYYellow** As Long

Description

The **CMYYellow** property returns or sets the yellow color in the CMY color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMY is a color mode made up of cyan (C), magenta (M), and yellow (Y). This mode is used in the three-color printing process. In Corel applications, the CMY mode is the inverse of the RGB mode, with values ranging from 0 to 255. The CMY color mode is based on the CMY color model.

{button ,AL(`CLS_Color;FNC_CMYYellow')} **Related Topics**

Color.Gray

Property **Gray** As Long

Description

The **Gray** property returns or sets the Gray value in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Grayscale color model is a color mode that displays images using 256 shades of gray. Each color is defined as a value between 0 and 255, where 0 is darkest (black) and 255 is lightest (white). In the RGB color mode, a grayscale value corresponds to equal amounts of all RGB colors; in CMYK, a grayscale value corresponds to zero C, M, and Y values, with a positive K value; in HSB, a grayscale value corresponds to zero H and S values, with a positive B value. The Grayscale color mode is based on the Grayscale color model.

{button ,AL(`CLS_Color;FNC_Gray')} **Related Topics**

Color.HLSHue

Property **HLSHue** As Long

Description

The **HLSHue** property assigns the Hue value for the HLS color model in Corel PHOTO-PAINT. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The HLS model is a variation of the HSB model and contains three components: hue, lightness, and saturation. Hue determines color (yellow, orange, red, etc.); lightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). The circular visual selector defines the H value (0 to 360) and the S value (0 to 100); the vertical visual selector defines the L value (0 to 100).

{button ,AL(`CLS_Color;FNC_HLSHue')} **Related Topics**

Color.HLSLightness

Property **HLSLightness** As Long

Description

The **HLSLightness** property assigns the Lightness value for the HLS color model in Corel PHOTO-PAINT. Lightness determines the intensity of the color. Values range from 0 to 100.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The HLS model is a variation of the HSB model and contains three components: hue, lightness, and saturation. Hue determines color (yellow, orange, red, etc.); lightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). The circular visual selector defines the H value (0 to 360) and the S value (0 to 100); the vertical visual selector defines the L value (0 to 100).

{button ,AL(`CLS_Color;FNC_HLSLightness')} **Related Topics**

Color.HLSSaturation

Property **HLSSaturation** As Long

Description

The **HLSSaturation** property assigns the Saturation value for the HLS color model in Corel PHOTO-PAINT. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The HLS model is a variation of the HSB model and contains three components: hue, lightness, and saturation. Hue determines color (yellow, orange, red, etc.); lightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). The circular visual selector defines the H value (0 to 360) and the S value (0 to 100); the vertical visual selector defines the L value (0 to 100).

{button ,AL(`CLS_Color;FNC_HLSSaturation')} **Related Topics**

Color.HSBBrightness

Property **HSBBrightness** As Long

Description

The **HSBBrightness** property assigns the Saturation value in the HSB color model in Corel PHOTO-PAINT. Brightness is the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

HSB is a color model that approximates the way the human eye perceives color. In the HSB model, color is defined by three components: hue, saturation, and brightness. Hue determines color (yellow, orange, red, etc.); brightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). In the HSB color model, Hue (H) is expressed as a degree of rotation on a circular color wheel. Saturation (S) and brightness (B) are expressed as percentages of full intensity.

{button ,AL(`CLS_Color;FNC_HSBBrightness')} **Related Topics**

Color.HSBHue

Property **HSBHue** As Long

Description

The **HSBHue** property assigns the Hue value in the HSB color model in Corel PHOTO-PAINT. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

HSB is a color model that approximates the way the human eye perceives color. In the HSB model, color is defined by three components: hue, saturation, and brightness. Hue determines color (yellow, orange, red, etc.); brightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). In the HSB color model, Hue (H) is expressed as a degree of rotation on a circular color wheel. Saturation (S) and brightness (B) are expressed as percentages of full intensity.

{button ,AL(`CLS_Color;FNC_HSBHue')} **Related Topics**

Color.HSBSaturation

Property **HSBSaturation** As Long

Description

The **HSBSaturation** property assigns the Saturation value in the HSB color model in Corel PHOTO-PAINT. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

HSB is a color model that approximates the way the human eye perceives color. In the HSB model, color is defined by three components: hue, saturation, and brightness. Hue determines color (yellow, orange, red, etc.); brightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). In the HSB color model, Hue (H) is expressed as a degree of rotation on a circular color wheel. Saturation (S) and brightness (B) are expressed as percentages of full intensity.

{button ,AL(`CLS_Color;FNC_HSBSaturation')} **Related Topics**

Color.LabComponentA

Property **LabComponentA** As Long

Description

The **LabComponentA** property returns or sets the A component in the LAB color model in Corel PHOTO-PAINT. This property sets one of two chromatic components in the LAB color model. "A" refers to a range of colors between green and red.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The LAB color model is a color model created by the Commission Internationale de l'Eclairage (CIE). It contains a luminance (or lightness) component (L) and two chromatic components: "a" (green to red) and "b" (blue to yellow). The Lab color mode is based on the Lab color model.

{button ,AL(`CLS_Color;FNC_LabComponentA')} **Related Topics**

Color.LabComponentB

Property **LabComponentB** As Long

Description

The **LabComponentB** property returns or sets the B component in the LAB color model in Corel PHOTO-PAINT. This property sets one of two chromatic components in the LAB color model. "B" refers to a range of colors between blue and yellow.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The LAB color model is a color model created by the Commission Internationale de l'Eclairage (CIE). It contains a luminance (or lightness) component (L) and two chromatic components: "a" (green to red) and "b" (blue to yellow). The Lab color mode is based on the Lab color model.

{button ,AL(`CLS_Color;FNC_LabComponentB')} **Related Topics**

Color.LabLuminance

Property **LabLuminance** As Long

Description

The **LabLuminance** property returns or sets the luminance level in the LAB color model in Corel PHOTO-PAINT. This property sets the Luminance, or brightness, value in the LAB color model. Values range from 0 to 255.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The LAB color model is a color model created by the Commission Internationale de l'Eclairage (CIE). It contains a luminance (or lightness) component (L) and two chromatic components: "a" (green to red) and "b" (blue to yellow). The Lab color mode is based on the Lab color model.

{button ,AL(`CLS_Color;FNC_LabLuminance')} **Related Topics**

Color.Name

Property **Name**([ByVal Components As Boolean = False]) As String

Description

The **Name** property returns a string value that identifies a color in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

This property returns a Read-Only value.

Parameters	Description
Components	The **Components** parameter sets a True or False value that indicates whether or not to return the color's components when using the Name property. This value is optional and the default value is False.

{button ,AL(`CLS_Color;FNC_Name')} **Related Topics**

Color.PaletteID

Property **PaletteID** As **cdrPaletteID**

Description

The **PaletteID** property returns a value associated with the type of a palette in Corel PHOTO-PAINT. The value returned is cdrPaletteID.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The **PaletteID** returns a Read-Only value.

{button ,AL(`CLS_Color;FNC_PaletteID')} **Related Topics**

Color.PaletteIndex

Property **PaletteIndex** As Long

Description

The **PaletteIndex** property returns or sets the index value of a color in a palette in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

{button ,AL(`CLS_Color;FNC_PaletteIndex')} **Related Topics**

Color.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Color;FNC_Parent')} **Related Topics**

Color.RGBBlue

Property **RGBBlue** As Long

Description

The **RGBBlue** property returns or sets the blue color value in the RGB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

RGB is a color mode that contains three components: red (R), green (G), and blue (B). The RGB color mode is based on the RGB color model. In the RGB color mode, a value between 0 and 255 is assigned to each channel of red, green, and blue. An RGB color with the component values 0:25:118, for example, contains no red, some green, and more blue, resulting in a slightly greenish blue color. Monitors, scanners, and the human eye use RGB to produce or detect color.

{button ,AL(`CLS_Color;FNC_RGBBlue')} **Related Topics**

Color.RGBGreen

Property **RGBGreen** As Long

Description

The **RGBGreen** property returns or sets the green color value in the RGB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

RGB is a color mode that contains three components: red (R), green (G), and blue (B). The RGB color mode is based on the RGB color model. In the RGB color mode, a value between 0 and 255 is assigned to each channel of red, green, and blue. An RGB color with the component values 0:25:118, for example, contains no red, some green, and more blue, resulting in a slightly greenish blue color. Monitors, scanners, and the human eye use RGB to produce or detect color.

{button ,AL(`CLS_Color;FNC_RGBGreen')} **Related Topics**

Color.RGBRed

Property **RGBRed** As Long

Description

The **RGBRed** property returns or sets the red color value in the RGB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

RGB is a color mode that contains three components: red (R), green (G), and blue (B). The RGB color mode is based on the RGB color model. In the RGB color mode, a value between 0 and 255 is assigned to each channel of red, green, and blue. An RGB color with the component values 0:25:118, for example, contains no red, some green, and more blue, resulting in a slightly greenish blue color. Monitors, scanners, and the human eye use RGB to produce or detect color.

{button ,AL(`CLS_Color;FNC_RGBRed')} **Related Topics**

Color.Tint

Property **Tint** As Long

Description

The **Tint** property returns or sets the tint level in a fixed palette's color in Corel PHOTO-PAINT. Tints are lighter shades of a spot color that are created by changing the percentage tint value.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_Color;FNC_Tint')} **Related Topics**

Color.Type

Property **Type** As **cdrColorType**

Description

The **Type** property returns or sets the color type in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The **Type** property returns cdrColorType.

{button ,AL(`CLS_Color;FNC_Type')} **Related Topics**

Color.YIQChromaI

Property **YIQChromaI** As Long

Description

The **YIQChromaII** returns or sets the I color value in the YIQ color model in Corel PHOTO-PAINT. The I value sets one of two chromaticity values in the YIQ color model. In monitor calibration, chromaticity refers to the chroma (hue adjustment) of your monitor. Values range from 0 to 255.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The YIQ color model is a color model used in television broadcast systems (North American video standard - NTSC). Colors are split into a luminance value (Y) and two chromaticity values (I and Q). On a color monitor, all three components are visible. On a monochrome monitor, only the Y component is visible. The square, two-dimensional visual selector defines the I and Q values, and the vertical visual selector defines the Y value. All values are scaled from 0 to 255.

{button ,AL(`CLS_Color;FNC_YIQChromaI')} **Related Topics**

Color.YIQChromaQ

Property **YIQChromaQ** As Long

Description

The **YIQChromaIQ** returns or sets the Q color value in the YIQ color model in Corel PHOTO-PAINT. The Q value sets one of two chromaticity values in the YIQ color model. In monitor calibration, chromaticity refers to the chroma (hue adjustment) of your monitor. Values range from 0 to 255.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The YIQ color model is a color model used in television broadcast systems (North American video standard - NTSC). Colors are split into a luminance value (Y) and two chromaticity values (I and Q). On a color monitor, all three components are visible. On a monochrome monitor, only the Y component is visible. The square, two-dimensional visual selector defines the I and Q values, and the vertical visual selector defines the Y value. All values are scaled from 0 to 255.

{button ,AL(`CLS_Color;FNC_YIQChromaQ')} **Related Topics**

Color.YIQLuminanceY

Property **YIQLuminanceY** As Long

Description

The **YIQLuminanceY** returns or sets the Y color value in the YIQ color model in Corel PHOTO-PAINT. The Y value sets the luminance, or brightness, in the YIQ color model. Values range from 0 to 255.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The YIQ color model is a color model used in television broadcast systems (North American video standard - NTSC). Colors are split into a luminance value (Y) and two chromaticity values (I and Q). On a color monitor, all three components are visible. On a monochrome monitor, only the Y component is visible. The square, two-dimensional visual selector defines the I and Q values, and the vertical visual selector defines the Y value. All values are scaled from 0 to 255.

{button ,AL(`CLS_Color;FNC_YIQLuminanceY')} **Related Topics**

Color.BWAssign

Sub **BWAssign**(ByVal **White** As Boolean)

Description

The **BWAssign** method sets True or False value that assigns the Black and White color model in Corel PHOTO-PAINT. With this method, True means white and False refers to black.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Black and White color model is a 1-bit color mode that stores images as two solid colors - black and white - with no gradations. This color mode is useful for line art and simple graphics.

Parameters	Description
White	Sets the appearance of white in the Black and White color model. This value is a True or False value.

{button ,AL(`CLS_Color;FNC_BWAssign')} **Related Topics**

Color.CMYAssign

Sub **CMYAssign**(ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long)

Description

The **CMYAssign** method assigns the CMY color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMY is a color mode made up of cyan (C), magenta (M), and yellow (Y). This mode is used in the three-color printing process. In Corel applications, the CMY mode is the inverse of the RGB mode, with values ranging from 0 to 255. The CMY color mode is based on the CMY color model.

Parameters	Description
Cyan	Sets the level of cyan in the CMY color model. Values range from 0 to 255.
Magenta	Sets the level of magenta in the CMY color model. Values range from 0 to 255.
Yellow	Sets the level of yellow in the CMY color model. Values range from 0 to 255.

{button ,AL(`CLS_Color;FNC_CMYAssign')} **Related Topics**

Color.CMYKAssign

Sub **CMYKAssign**(ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long, ByVal **Black** As Long)

Description

The **CMYKAssign** property assigns the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

Parameters	Description
Cyan	Sets the level of cyan in the CMYK color model. Values are percentages and range from 0 to 100.
Magenta	Sets the level of magenta in the CMYK color model. Values are percentages and range from 0 to 100.
Yellow	Sets the level of yellow in the CMYK color model. Values are percentages and range from 0 to 100.
Black	Sets the level of black in the CMYK color model. Values are percentages and range from 0 to 100.

{button ,AL(`CLS_Color;FNC_CMYKAssign')} **Related Topics**

Color.ConvertToBW

Sub **ConvertToBW**()

Description

The **ConvertToBW** method converts the active color model to the Black and White color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Black and White color model is a 1-bit color mode that stores images as two solid colors - black and white - with no gradations. This color mode is useful for line art and simple graphics.

{button ,AL(`CLS_Color;FNC_ConvertToBW')} **Related Topics**

Color.ConvertToCMY

Sub **ConvertToCMY**()

Description

The **ConvertToCMY** method converts the active color model to the CMY color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMY is a color mode made up of cyan (C), magenta (M), and yellow (Y). This mode is used in the three-color printing process. In Corel applications, the CMY mode is the inverse of the RGB mode, with values ranging from 0 to 255. The CMY color mode is based on the CMY color model.

{button ,AL(`CLS_Color;FNC_ConvertToCMY')} **Related Topics**

Color.ConvertToCMYK

Sub **ConvertToCMYK**()

Description

The **ConvertToCMYK** method converts the active color model to the CMYK color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

CMYK is a color mode made up of cyan (C), magenta (M), yellow (Y), and black (K). In the CMYK color mode, color values are expressed as percentages, so a value of 100 for an ink means that it is applied at full saturation. Used in most full-color commercial printing, CMYK is like CMY, but the addition of black (K) allows for true blacks and a wider tonal range. The CMYK color mode is based on the CMYK color model.

{button ,AL(`CLS_Color;FNC_ConvertToCMYK')} **Related Topics**

Color.ConvertToFixed

Sub **ConvertToFixed**(ByVal **PaletteID** As **cdrPaletteID**)

Description

The **ConvertToFixed** method converts the active color model to a color in the fixed palette of Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

Parameters	Description
PaletteID	Sets the fixed palette in Corel PHOTO-PAINT. This parameter returns cdrPaletteID.

{button ,AL(`CLS_Color;FNC_ConvertToFixed')} **Related Topics**

Color.ConvertToGray

Sub **ConvertToGray**()

Description

The **ConvertToGray** method converts the active color model to the Grayscale color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Grayscale color model is a color mode that displays images using 256 shades of gray. Each color is defined as a value between 0 and 255, where 0 is darkest (black) and 255 is lightest (white). In the RGB color mode, a grayscale value corresponds to equal amounts of all RGB colors; in CMYK, a grayscale value corresponds to zero C, M, and Y values, with a positive K value; in HSB, a grayscale value corresponds to zero H and S values, with a positive B value. The Grayscale color mode is based on the Grayscale color model.

{button ,AL(`CLS_Color;FNC_ConvertToGray')} **Related Topics**

Color.ConvertToHLS

Sub **ConvertToHLS**()

Description

The **ConvertToHLS** method converts the active color model to the HLS color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The HLS model is a variation of the HSB model and contains three components: hue, lightness, and saturation. Hue determines color (yellow, orange, red, etc.); lightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). The circular visual selector defines the H value (0 to 360) and the S value (0 to 100); the vertical visual selector defines the L value (0 to 100).

{button ,AL(`CLS_Color;FNC_ConvertToHLS')} **Related Topics**

Color.ConvertToHSB

Sub **ConvertToHSB**()

Description

The **ConvertToHSB** method converts the active color model to the HSB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

HSB is a color model that approximates the way the human eye perceives color. In the HSB model, color is defined by three components: hue, saturation, and brightness. Hue determines color (yellow, orange, red, etc.); brightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). In the HSB color model, Hue (H) is expressed as a degree of rotation on a circular color wheel. Saturation (S) and brightness (B) are expressed as percentages of full intensity.

{button ,AL(`CLS_Color;FNC_ConvertToHSB')} **Related Topics**

Color.ConvertToLab

Sub **ConvertToLab**()

Description

The **ConvertToLab** method converts the active color model to the LAB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The LAB color model is a color model created by the Commission Internationale de l'Eclairage (CIE). It contains a luminance (or lightness) component (L) and two chromatic components: "a" (green to red) and "b" (blue to yellow). The Lab color mode is based on the Lab color model.

{button ,AL(`CLS_Color;FNC_ConvertToLab')} **Related Topics**

Color.ConvertToRGB

Sub **ConvertToRGB**()

Description

The **ConvertToRGB** method converts the active color model to the RGB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

RGB is a color mode that contains three components: red (R), green (G), and blue (B). The RGB color mode is based on the RGB color model. In the RGB color mode, a value between 0 and 255 is assigned to each channel of red, green, and blue. An RGB color with the component values 0:25:118, for example, contains no red, some green, and more blue, resulting in a slightly greenish blue color. Monitors, scanners, and the human eye use RGB to produce or detect color.

Corel PHOTO-PAINT lets you convert an image to the following color modes: Black-and-White, Grayscale, Duotone, RGB, CMYK, or LAB. You can also convert a bitmap to the Paletted color mode.

{button ,AL(`CLS_Color;FNC_ConvertToRGB')} **Related Topics**

Color.ConvertToYIQ

Sub **ConvertToYIQ**()

Description

The **ConvertToYIQ** method converts the active color model to the YIQ color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The YIQ color model is a color model used in television broadcast systems (North American video standard - NTSC). Colors are split into a luminance value (Y) and two chromaticity values (I and Q). On a color monitor, all three components are visible. On a monochrome monitor, only the Y component is visible. The square, two-dimensional visual selector defines the I and Q values, and the vertical visual selector defines the Y value. All values are scaled from 0 to 255.

{button ,AL(`CLS_Color;FNC_ConvertToYIQ')} **Related Topics**

Color.CopyAssign

Sub **CopyAssign**(ByVal **Color** As **Color**)

Description

The **CopyAssign** method copies the color model properties from a color object and applies them to the active color in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

Parameters	Description
Color	The **Color** parameter identifies the color who's color model properties are copied with the CopyAssign method.

{button ,AL(`CLS_Color;FNC_CopyAssign')} **Related Topics**

Color.CorelScriptAssign

Sub **CorelScriptAssign**(ByVal **ColorModel** As Long, ByVal **V1** As Long, [ByVal V2 As Long = 0], [ByVal V3 As Long = 0], [ByVal V4 As Long = 0], [ByVal V5 As Long = 0], [ByVal V6 As Long = 0], [ByVal V7 As Long = 0])

Description

The **CorelScriptAssign** method assigns a color object based on Corel SCRIPT color specification values in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

Parameters	Description
ColorModel	The **ColorModel** parameter identifies the color model used to create the new color. This parameter specifies the numeric variable that is assigned to the color model. A color model is a simple color chart that defines the range of colors displayed in a color mode.
V1	The **V1** parameter specifies the numeric variable that is assigned to the first color component of the selected color model. For example, cyan is the first color component of the CMYK color model. The value is a percentage and can range from 1 to 100. A color that is selected at 100 is applied with full saturation.
V2	The V2 parameter specifies the numeric variable that is assigned to the second color component of the selected color model. This value is optional and the default value is 0.
V3	The **V3** parameter specifies the numeric variable that is assigned to the third color component of the selected color model. This value is optional and the default value is 0.
V4	The **V4** parameter specifies the numeric variable that is assigned to the fourth color component of the selected color model. This value is optional and the default value is 0.
V5	The **V5** parameter specifies the numeric variable that is assigned to the fifth color component of the selected color model. This value is optional and the default value is 0.
V6	The **V6** parameter specifies the numeric variable that is assigned to the sixth color component of the selected color model. This value is optional and the default value is 0.
V7	The **V7** parameter specifies the numeric variable that is assigned to the seventh color component of the selected color model. This value is optional and the default value is 0.

{button ,AL(`CLS_Color;FNC_CorelScriptAssign')} **Related Topics**

Color.CorelScriptGetComponent

Sub **CorelScriptGetComponent**(ByRef **ColorModel** As Long, ByRef **V1** As Long, [ByRef V2 As Long = 0], [ByRef V3 As Long = 0], [ByRef V4 As Long = 0], [ByRef V5 As Long = 0], [ByRef V6 As Long = 0], [ByRef V7 As Long = 0])

Description

The **CorelScriptGetComponent** method gets Corel SCRIPT values from a color object for use in Corel SCRIPT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

Parameters	Description
ColorModel	The **ColorModel** parameter identifies the color model used to create the new color. This parameter specifies the numeric variable that is assigned to the color model. A color model is a simple color chart that defines the range of colors displayed in a color mode.
V1	The **V1** parameter specifies the numeric variable that is assigned to the first color component of the selected color model. For example, cyan is the first color component of the CMYK color model. The value is a percentage and can range from 1 to 100. A color that is selected at 100 is applied with full saturation.
V2	The V2 parameter specifies the numeric variable that is assigned to the second color component of the selected color model. This value is optional and the default value is 0.
V3	The **V3** parameter specifies the numeric variable that is assigned to the third color component of the selected color model. This value is optional and the default value is 0.
V4	The **V4** parameter specifies the numeric variable that is assigned to the fourth color component of the selected color model. This value is optional and the default value is 0.
V5	The **V5** parameter specifies the numeric variable that is assigned to the fifth color component of the selected color model. This value is optional and the default value is 0.
V6	The **V6** parameter specifies the numeric variable that is assigned to the sixth color component of the selected color model. This value is optional and the default value is 0.
V7	The **V7** parameter specifies the numeric variable that is assigned to the seventh color component of the selected color model. This value is optional and the default value is 0.

{button ,AL(`CLS_Color;FNC_CorelScriptGetComponent')} **Related Topics**

Color.FixedAssign

Sub **FixedAssign**(ByVal **PaletteID** As **cdrPaletteID**, ByVal **PaletteIndex** As Long, [ByVal Tint As Long = 100])

Description

The **FixedAssign** method assigns a fixed palette color in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Parameters	Description
PaletteID	Sets the Palette ID that uniquely identifies which color palette to use in Corel PHOTO-PAINT.
PaletteIndex	Sets the color to use by referencing the index number of a color within a palette.
Tint	Sets the tint of the palette color. Tints are lighter shades of a spot color that are created by changing the percentage tint value. This value is optional and the default value is 100.

{button ,AL(`CLS_Color;FNC_FixedAssign')} **Related Topics**

Color.GrayAssign

Sub **GrayAssign**(ByVal **GrayValue** As Long)

Description

The **GrayAssign** method assigns the Grayscale color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The Grayscale color model is a color mode that displays images using 256 shades of gray. Each color is defined as a value between 0 and 255, where 0 is darkest (black) and 255 is lightest (white). In the RGB color mode, a grayscale value corresponds to equal amounts of all RGB colors; in CMYK, a grayscale value corresponds to zero C, M, and Y values, with a positive K value; in HSB, a grayscale value corresponds to zero H and S values, with a positive B value. The Grayscale color mode is based on the Grayscale color model.

Parameters	Description
GrayValue	Sets the level of gray in the Grayscale color model. Each color is defined as a value between 0 and 255, where 0 is darkest (black) and 255 is lightest (white).

{button ,AL(`CLS_Color;FNC_GrayAssign')} **Related Topics**

Color.HLSAssign

Sub **HLSAssign**(ByVal **Hue** As Long, ByVal **Lightness** As Long, ByVal **Saturation** As Long)

Description

The **HLSAssign** method assigns the HLS color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The HLS model is a variation of the HSB model and contains three components: hue, lightness, and saturation. Hue determines color (yellow, orange, red, etc.); lightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). The circular visual selector defines the H value (0 to 360) and the S value (0 to 100); the vertical visual selector defines the L value (0 to 100).

Parameters	Description
Hue	Sets the Hue for the HLS color model. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.
Lightness	Sets the Lightness for the HLS color model. Lightness determines the intensity of the color. Values range from 0 to 100.
Saturation	Sets the Saturation for the HLS color model. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.

{button ,AL(`CLS_Color;FNC_HLSAssign')} **Related Topics**

Color.HSBAssign

Sub **HSBAssign**(ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Description

The **HSBAssign** method assigns the HSB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

HSB is a color model that approximates the way the human eye perceives color. In the HSB model, color is defined by three components: hue, saturation, and brightness. Hue determines color (yellow, orange, red, etc.); brightness determines perceived intensity (lighter or darker color); and saturation determines color depth (from dull to intense). In the HSB color model, Hue (H) is expressed as a degree of rotation on a circular color wheel. Saturation (S) and brightness (B) are expressed as percentages of full intensity.

Parameters	Description
Hue	Sets the Hue for the HSB color model. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.
Saturation	Sets the Saturation for the HSB color model. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.
Brightness	Sets the Brightness for the HSB color model. Brightness is the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

{button ,AL(`CLS_Color;FNC_HSBAssign')} **Related Topics**

Color.LabAssign

Sub **LabAssign**(ByVal **L** As Long, ByVal **A** As Long, ByVal **B** As Long)

Description

The **LabAssign** method assigns the LAB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The LAB color model is a color model created by the Commission Internationale de l'Eclairage (CIE). It contains a luminance (or lightness) component (L) and two chromatic components: "a" (green to red) and "b" (blue to yellow). The Lab color mode is based on the Lab color model.

Parameters	Description
L	Sets the Luminance, or brightness, value in the LAB color model. Values range from 0 to 255.
A	Sets one of two chromatic components in the LAB color model. "A" refers to a range of colors between green and red.
B	Sets one of two chromatic components in the LAB color model. "B" refers to a range of colors between blue and yellow.

{button ,AL(`CLS_Color;FNC_LabAssign')} **Related Topics**

Color.RegistrationAssign

Sub **RegistrationAssign**()

Description

The **RegistrationAssign** method sssigns the Registration color model that allows color trapping in Corel PHOTO-PAINT. Color trapping is necessary to compensate for poor color registration that occurs when the printing plates used to print each color, called color separations, are not aligned perfectly. Poor registration causes unintentional white slivers to appear between adjoining colors. Trapping is accomplished by intentionally overlapping colors so that minor problems with alignment are not noticed.

Color trapping is achieved by overprinting. Usually, portions of an object that are obscured by another object are not printed. However, if the top object is set to overprint, the obscured portions of any underlying objects print anyway, causing an overlap. This makes white gaps between different colors unlikely. Overprinting works best when the top color is much darker than the underlying color; otherwise, an undesirable third color may result (for example, red over yellow may result in an orange object).

{button ,AL(`CLS_Color;FNC_RegistrationAssign')} **Related Topics**

Color.RGBAssign

Sub **RGBAssign**(ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long)

Description

The **RGBAssign** method assigns the RGB color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

RGB is a color mode that contains three components: red (R), green (G), and blue (B). The RGB color mode is based on the RGB color model. In the RGB color mode, a value between 0 and 255 is assigned to each channel of red, green, and blue. An RGB color with the component values 0:25:118, for example, contains no red, some green, and more blue, resulting in a slightly greenish blue color. Monitors, scanners, and the human eye use RGB to produce or detect color.

Parameters	Description
Red	Sets the level of red in the RGB color model. Values range from 0 to 255.
Green	Sets the level of green in the RGB color model. Values range from 0 to 255.
Blue	Sets the level of blue in the RGB color model. Values range from 0 to 255.

{button ,AL(`CLS_Color;FNC_RGBAssign')} **Related Topics**

Color.UserAssign

Sub **UserAssign**()

Description

The **UserAssign** method brings up the Color dialog box, allowing the user to assign a color in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_Color;FNC_UserAssign')} **Related Topics**

Color.UserAssignEx

Function **UserAssignEx**() As Boolean

Description

The **UserAssignEx** method sets a True or False value that brings up the Color dialog box, allowing the user to select a color in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_Color;FNC_UserAssignEx')} **Related Topics**

Color.YIQAssign

Sub **YIQAssign**(ByVal **Y** As Long, ByVal **I** As Long, ByVal **Q** As Long)

Description

The **YIQAssign** method assigns the YIQ color model in Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The YIQ color model is a color model used in television broadcast systems (North American video standard - NTSC). Colors are split into a luminance value (Y) and two chromaticity values (I and Q). On a color monitor, all three components are visible. On a monochrome monitor, only the Y component is visible. The square, two-dimensional visual selector defines the I and Q values, and the vertical visual selector defines the Y value. All values are scaled from 0 to 255.

Parameters	Description
Y	Sets the luminance, or brightness, value in the YIQ color model. Values range from 0 to 255.
I	Sets one of two chromaticity values in the YIQ color model. In monitor calibration, chromaticity refers to the chroma (hue adjustment) of your monitor. Values range from 0 to 255.
Q	Sets one of two chromaticity values in the YIQ color model. In monitor calibration, chromaticity refers to the chroma (hue adjustment) of your monitor. Values range from 0 to 255.

{button ,AL(`CLS_Color;FNC_YIQAssign')} **Related Topics**

Without any light or a viewer, objects all around us are colorless. Color only occurs in our minds after our visual sensory system has seen the wavelengths that give objects their color. Based on how people perceive color, the HSB color model defines color in three attributes:

- Hue (H)
- Saturation (S)
- Brightness (B)

Hue (H) is the name we give a color in everyday language. Hues form the Color Wheel. The hue of a lemon is yellow, that of a strawberry is red. Saturation (S) refers to vividness of the color or how much color concentration does the object contain. The figurine does not contain very much yellow when compared to the yellow saturation of lemon. Colors can be separated into bright or dark colors when their Brightness (B) is compared. Brightness refers to adding or removing whiteness from a color. The mask is bright and lighter then the dark yellow lemon.

cdrPantone=1

cdrCMYK=2

cdrCMY=4

cdrRGB=5

cdrHSB=6

cdrHLS=7

cdrBlackAndWhite=8

cdrGray=9

cdrColorPantone=1

cdrColorCMYK=2

cdrColorCMY=4

cdrColorRGB=5

cdrColorHSB=6

cdrColorHLS=7

cdrColorBlackAndWhite=8

cdrColorGray=9

cdrColorYIQ=11

cdrColorLab=12

cdrColorPantoneHex=14

cdrColorRegistration=20

cdrColorSpot=25

cdrColorMixed=99

A great deal of color research has been accomplished in order to acquire a color model that is device independent and repeatable. In 1931 La Commision Internationale de L'Eclairage (CIE) defined a device-independent color model, based on how the human eye perceives color. The CIE Lab model incorporates the theory that a color cannot be both green and red at the same time nor can it be yellow and blue at the same time. As such, single values are used to describe the green/red and blue/yellow components of any color. Lab stands for the three values this model uses to define colora lightness value (L) which can range from 0 to 100 and two chromaticity ranges: green to red (a) and blue to yellow (b). The two chromaticity values can range from +120 to -120. Lab (sometimes called L*a*b*) provides a system for defining color that bases color values on widely accepted standards rather than on individual color-producing devices.

The millions of colors your monitor produces can all be described as amounts of red, green, and blue. These three color components form the basis for the RGB (Red, Green, and Blue) color model. Each of the three colors is assigned a numeric value between 0 and 255. The RGB model is based on colors of light, and higher RGB values correspond to the presence of greater quantities of white light. Consequently, higher RGB values result in lighter colors. When all three color components are at the maximum value, the resulting color is white light. Because the RGB model creates colors by adding light, it is called an additive color model. Monitors and scanners can employ the additive color model because they emit light. They emit particles of red, green, and blue light and create the illusion of millions of different colors.

One of the limitations of the RGB model is that it is device dependent. This means that not only are there color variations between monitors and scanners by different manufacturers but there are color variations between identical devices from the same manufacturer. All monitors drift over time and display colors differently making it imperative to regularly calibrate your monitor and the other electronic devices you use to create your projects. The RGB model cannot be a color standard because its color results are not 100 percent repeatable.

cdrTRUMATCH=1

cdrPANTONEProcess=2

cdrPANTONECorel8=3

cdrUniform=7

cdrFOCOLTONE=8

cdrSpectraMaster=9

cdrTOYO=10

cdrDIC=11

cdrPANTONEHexCoated=12

cdrPANTONEHexUncoated=24

cdrLab=13

cdrNetscapeNavigator=14

cdrInternetExplorer=15

cdrPANTONECoated=17

cdrPANTONEUncoated=18

cdrPANTONEMetallic=20

cdrPANTONEPastelCoated=21

cdrPANTONEPastelUncoated=22

cdrHKS=23
cdrCustom=0

ColorMask properties

ColorMask Legend

- Application
  - BrightnessModel
- ColorMaskColors
  - Inverted
  - Mode
  - OutOfGamut
- Parent
  - Smoothing
  - Threshold

ColorMask methods

ColorMask Legend

- AddColor
- Reset

ColorMask

Class **ColorMask**

Properties Methods Referenced By

The **ColorMask** class defines the characteristics of color mask objects and describes the look and behavior of the collection objects through its properties and methods.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

{button ,AL(`CLS_ColorMask')} **Related Topics**

ColorMask.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_ColorMask;FNC_Application')} **Related Topics**

ColorMask.BrightnessModel

Property **BrightnessModel** As **pntMaskBrightnessModel**

Description

The **BrightnessModel** property returns or sets the mask brightness model of a color mask object in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

Brightness refers to the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

This value returns pntMaskBrightnessModel.

{button ,AL(`CLS_ColorMask;FNC_BrightnessModel')} **Related Topics**

pntMaskToWhite	0
pntMaskToBlack	1

ColorMask.ColorMaskColors

Property **ColorMaskColors** As **ColorMaskColors**

Description

The **ColorMaskColors** property returns a value associated with the **ColorMaskColors** collection in Corel PHOTO-PAINT. This collection contains information about the colors used in color masks in Corel PHOTO-PAINT.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

This property returns a Read-Only value.

{button ,AL(`CLS_ColorMask;FNC_ColorMaskColors')} **Related Topics**

ColorMask.Inverted

Property **Inverted** As Boolean

Description

The **Inverted** property returns or sets a True of False value that indicates whether or not a color mask is inverted in Corel PHOTO-PAINT. Often, it is easier to select the areas you want to protect from changes and then invert the selection. For example, if you want to edit a complex, irregularly shaped area, you can begin by selecting the surrounding area. Then, you can reverse the mask so that the area that was initially selected is protected, and the area that was masked is selected, i.e., available for editing.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

{button ,AL(`CLS_ColorMask;FNC_Inverted')} **Related Topics**

ColorMask.Mode

Property **Mode** As **pntColorMaskMode**

Description

The **Mode** property returns or sets the color mask mode of a color mask in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

This value returns pntColorMaskMode.

{button ,AL(`CLS_ColorMask;FNC_Mode')} **Related Topics**

pntColorMaskNormal	0
pntColorMaskHue	1
pntColorMaskSaturation	2
pntColorMaskBrightness	4

ColorMask.OutOfGamut

Property **OutOfGamut** As Boolean

Description

The **OutOfGamut** property returns or sets a True or False value that indicates whether or not the color used in a color mask is an out of gamut color. An out of gamut color is a color that is beyond the capabilities (outside the gamut) of a device. When you choose an out-of-gamut color, an in-gamut color button appears beside the color swatch in the Color window.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

{button ,AL(`CLS_ColorMask;FNC_OutOfGamut')} **Related Topics**

ColorMask.Parent

Property **Parent** As **Application**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_ColorMask;FNC_Parent')} **Related Topics**

ColorMask.Smoothing

Property **Smoothing** As Long

Description

The **Smoothing** property returns or sets a value that smoothes the edges of a color mask in Corel PHOTO-PAINT. You can smooth the edges of a selection by toning down the contrast between pixels on its edge. The smoothing radius value you specify determines the intensity of the smoothing effect. Smoothing is most useful when you work with complex color selections, which often have uneven edges. When you smooth the edges of a selection, masked areas that are completely surrounded by the selection might be removed.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

{button ,AL(`CLS_ColorMask;FNC_Smoothing')} **Related Topics**

ColorMask.Threshold

Property **Threshold** As Long

Description

The **Threshold** property returns or sets the threshold value for a color mask in Corel PHOTO-PAINT. Threshold is a level of tolerance for tonal variation in a bitmap image. For example, when you convert an image to the Black-and-White color mode, the threshold you set determines how many tonal values are converted to black and how many to white. Threshold settings are also used in color-sensitive masks and some Effects filters.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

{button ,AL(`CLS_ColorMask;FNC_Threshold')} **Related Topics**

ColorMask.AddColor

Sub **AddColor**(ByVal **Color** As **Color**, [ByVal Normal As Long = 0], [ByVal Hue As Long = 0], [ByVal Saturation As Long = 0], [ByVal Brightness As Long = 0])

Description

The **AddColor** method adds a color to a color mask in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

Parameters	Description
Color	Sets the color to be added to the color mask.
Normal	Sets the normal level of the new color. This value is optional and the default value is 0.
Hue	Sets the hue level of the new color. This value is optional and the default value is 0. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.
Saturation	Sets the saturation level of the new color. This value is optional and the default value is 0. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.
Brightness	Sets the brightness level of the new color. This value is optional and the default value is 0. Brightness is the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

{button ,AL(`CLS_ColorMask;FNC_AddColor')} **Related Topics**

pntColorMaskSampledColors	0
pntColorMaskReds	1
pntColorMaskYellows	2
pntColorMaskGreens	3
pntColorMaskCyans	4
pntColorMaskBlues	5
pntColorMaskMagentas	6
pntColorMaskHighlights	7
pntColorMaskMidtones	8
pntColorMaskShadows	9
pntColorMaskOutOfGamut	10

ColorMask.Reset

Sub **Reset**([ByVal Type As **pntColorMaskType** = pntColorMaskSampledColors (0)], [ByVal Smoothing As Long = 0], [ByVal Inverted As Boolean = False], [ByVal Mode As **pntColorMaskMode** = pntColorMaskNormal (0)], [ByVal Threshold As Long = 127], [ByVal BrightnessModel As **pntMaskBrightnessModel** = pntMaskToBlack (1)])

Description

The **Reset** method resets the values of a color mask in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

Parameters	Description
Type	Sets the color mask type. This value returns pntColorMaskType. This value is optional and the default value is **pntColorMaskSampledColors** (0)
Smoothing	Sets the smoothing value. This value is optional and the default value is 0. You can smooth the edges of a selection by toning down the contrast between pixels on its edge. The smoothing radius value you specify determines the intensity of the smoothing effect. Smoothing is most useful when you work with complex color selections, which often have uneven edges. When you smooth the edges of a selection, masked areas that are completely surrounded by the selection might be removed.
Inverted	Sets a True or False value that indicates whether or not to invert a color mask. This value is optional and the default value is False.
Mode	Sets the mask mode. This value returns pntColorMaskMode. This value is optional and the default value is **pntColorMaskNormal** (0)
Threshold	Sets the threshold level. This value is optional and the default value is 127. Threshold is a level of tolerance for tonal variation in a bitmap image. For example, when you convert an image to the Black-and-White color mode, the threshold you set determines how many tonal values are converted to black and how many to white. Threshold settings are also used in color-sensitive masks and some Effects filters.
BrightnessModel	Sets the brightness model type. This value returns pntMaskBrightnessModel. This value is optional and the default value is **pntMaskToBlack** (1)

{button ,AL(`CLS_ColorMask;FNC_Reset')} **Related Topics**

ColorMaskColor properties

ColorMaskColor Legend

- Application
  - Brightness
  - Color
  - Hue
  - Normal
- Parent
  - Saturation

ColorMaskColor methods

ColorMaskColor Legend

Delete

ColorMaskColor

Class **ColorMaskColor**

Properties Methods Referenced By

The **ColorMaskColor** class defines the characteristics of a color in a color mask object and describes the look and behavior of the collection objects through its properties and methods.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor')} **Related Topics**

ColorMaskColor.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_ColorMaskColor;FNC_Application')} **Related Topics**

ColorMaskColor.Brightness

Property **Brightness** As Long

Description

The **Brightness** property returns or sets the brightness level of a color in a color mask in Corel PHOTO-PAINT. Brightness is the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor;FNC_Brightness')} **Related Topics**

ColorMaskColor.Color

Property **Color** As **Color**

Description

The **Color** property returns or sets a value associated with a color in a color mask in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor;FNC_Color')} **Related Topics**

ColorMaskColor.Hue

Property **Hue** As Long

Description

The **Hue** property returns or sets the hue level of a color in a color mask in Corel PHOTO-PAINT. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor;FNC_Hue')} **Related Topics**

ColorMaskColor.Normal

Property **Normal** As Long

Description

The **Normal** property returns or sets the normal level of a color in a color mask in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor;FNC_Normal')} **Related Topics**

ColorMaskColor.Parent

Property **Parent** As **Application**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_ColorMaskColor;FNC_Parent')} **Related Topics**

ColorMaskColor.Saturation

Property **Saturation** As Long

Description

The **Saturation** property returns or sets the saturation level of a color in a color mask in Corel PHOTO-PAINT. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor;FNC_Saturation')} **Related Topics**

ColorMaskColor.Delete

Sub **Delete**()

Description

The **Delete** method deletes a color in a color mask object in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColor;FNC_Delete')} **Related Topics**

ColorMaskColors properties

ColorMaskColors Legend

- Application

- Count

-
- Item

- Parent

ColorMaskColors methods

ColorMaskColors Legend

Add

ColorMaskColors

Class **ColorMaskColors**

Properties Methods Referenced By

The **ColorMaskColors** class defines the characteristics of a **ColorMaskColors** collection objects and describes the look and behavior of the collection objects through its properties and methods.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

{button ,AL(`CLS_ColorMaskColors')} **Related Topics**

ColorMaskColors.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_ColorMaskColors;FNC_Application')} **Related Topics**

ColorMaskColors.Count

Property **Count** As Long

Description

The **Count** property counts the number of color mask colors in the **ColorMaskColors** collection of Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

This property returns a Read-Only value.

{button ,AL(`CLS_ColorMaskColors;FNC_Count')} **Related Topics**

ColorMaskColors.Item

Property **Item**(ByVal **Index** As Long) As **ColorMaskColor**

Description

The **Item** property returns a value associated with the index number of a color in the **ColorMaskColors** collection of Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

`ColorMaskColors(2)` refers to the second color mask color in the collection.

The **Item** property returns a Read-Only value. The **Item** property is the default property and may be omitted when referencing items in the collection.

Parameters	Description
Index	**Index** is a preset placeholder for each color mask color in the **ColorMaskColors** collection; it uniquely identifies each member of the collection.

{button ,AL(`CLS_ColorMaskColors;FNC_Item')} **Related Topics**

ColorMaskColors.Parent

Property **Parent** As **Application**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_ColorMaskColors;FNC_Parent')} **Related Topics**

ColorMaskColors.Add

Sub **Add**(ByVal **pVal** As **ColorMaskColor**)

Description

The **Add** method adds a color mask color to the **ColorMaskColors** collection in Corel PHOTO-PAINT. A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

Parameters	Description
pVal	Specifies the color mask color that is added to the **ColorMaskColors** collection.

{button ,AL(`CLS_ColorMaskColors;FNC_Add')} **Related Topics**

Colors properties

Colors Legend

- Application
- Count
-
- Item
- Parent

Colors

Class **Colors**

Properties Referenced By

The **Colors** class defines the characteristics of **Colors** collection objects and describes the look and behavior of the collection objects through its properties and methods.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

There are many different color models that define colors - HSB, RGB, CMYK, and CIE Lab color models. The RGB and CMYK color models are only two of a number of models developed to suit a variety of digital design and desktop publishing applications. It is not necessary to be familiar with all these models, but it is helpful to be familiar with a few of the more widely used ones.

We all see color differently. Color is subjective to the human eye. Each device that interacts with your project's file: the scanner, monitor, and printer may have a different color space. For example, a color that is visible to the human eye may not be reproducible by your printer.

Because there are so many color variations, a precise method for defining each color is required. For example, once you find the perfect shade of light orange, you need to be able to reproduce that color and possibly tell others how to do the same. A color model defines that perfect shade of light orange by breaking it down into precise components that allow you to accurately transmit the information to other people and to the electronic devices you use to create projects.

{button ,AL(`CLS_Colors')} **Related Topics**

Colors.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, **CorelScript** and **Application.CorelScript** can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub ColorsApp()
With ActivePalette.Colors
    MsgBox "You are using Corel PHOTO-PAINT " & .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Colors;FNC_Application')} **Related Topics**

Colors.Count

Property **Count** As Long

Description

The **Count** property returns the number of colors in the **Colors** collection of Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

The **Count** property returns a Read-Only value.

Example

The following code example displays the number of colors in the active palette's **Colors** collection in a message box:

```
Sub ColorsCount()
With ActivePalette.Colors
    MsgBox .Count
End With
End Sub
```

{button ,AL(`CLS_Colors;FNC_Count')} **Related Topics**

Colors.Item

Property **Item**(ByVal **Index** As Long) As **Color**

Description

The **Item** property returns a value associated with the index number of a color in the **Colors** collection of Corel PHOTO-PAINT. A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

`Colors(2)` refers to the second color in the **Colors** collection of Corel PHOTO-PAINT.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Colors.Item(2)` is the same as `Colors(2)` - they both reference the second color in the **Colors** collection.

You must reference a color in the collection by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each color in a **Colors** collection; it uniquely identifies each member of the collection.

Example

The following code example displays the color type of the **Colors** collection's third color in a message box:

```
Sub ColorsItem()
With ActivePalette.Colors
    MsgBox .Item(3).Type
End With
End Sub
```

{button ,AL(`CLS_Colors;FNC_Item')} **Related Topics**

Colors.Parent

Property **Parent** As **Palette**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the name of the **Colors** collection's parent palette in a message box:

```
Sub ColorsParent()
With ActivePalette.Colors
    MsgBox .Parent.Name
End With
End Sub
```

{button ,AL(`CLS_Colors;FNC_Parent')} **Related Topics**

ColorTable properties

ColorTable Legend

- Application
- Count
- Item
- Parent

ColorTable methods

ColorTable Legend

- AddColor
- GetIndexOfColor
- LoadPalette
- LoadPaletteFromFile
- Sort

ColorTable

Class **ColorTable**

Properties Methods Referenced By

The **ColorTable** class defines the characteristics of color table objects and describes the look and behavior of the collection objects through its properties and methods. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

A color is an effect applied to an object that alters the object's appearance by the way it reflects light. A color model is a system that defines the number and kind of colors that make up a bitmap image. Black-and-White, Grayscale, RGB, CMYK, and Paletted are examples of popular color modes. A color model is a system used to organize and define colors according to a set of basic properties that may be reproduced.

After you convert an image to the Paletted color mode, you can use the Color Table to customize the Color Palette of the image. A custom Color Palette is a collection of colors saved in a Color Palette file format. They are useful when you often use the same colors or when you want to work with a set of complementary colors.

{button ,AL(`CLS_ColorTable')} **Related Topics**

ColorTable.Application

Property **Application** As **Application**

Description

The **Parent** property returns a value associated with the parent object of a color table in Corel PHOTO-PAINT. A parent is an object that retains its shape but contains the color or texture of the child objects that are clipped to it in a clipping group. An object is always the parent to the objects that are listed above it in the Objects Docker window.

A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

This property returns a Read-Only value.

{button ,AL(`CLS_ColorTable;FNC_Application')} **Related Topics**

ColorTable.Count

Property **Count** As Long

Description

The **Count** property counts the number of color tables in the Corel PHOTO-PAINT. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

This property returns a Read-Only value.

{button ,AL(`CLS_ColorTable;FNC_Count')} **Related Topics**

ColorTable.Item

Property **Item**(ByVal **Index** As Long) As **Color**

Description

The **Item** property returns a value associated with the index number of a color table in Corel PHOTO-PAINT. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

The **Item** property returns a Read-Only value. You must reference an arrowhead in the collection by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each color table object in Corel PHOTO-PAINT; it uniquely identifies each color table.

{button ,AL(`CLS_ColorTable;FNC_Item')} **Related Topics**

ColorTable.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the parent object of a color table in Corel PHOTO-PAINT. A parent is an object that retains its shape but contains the color or texture of the child objects that are clipped to it in a clipping group. An object is always the parent to the objects that are listed above it in the Objects Docker window.

A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

This property returns a Read-Only value.

{button ,AL(`CLS_ColorTable;FNC_Parent')} **Related Topics**

ColorTable.AddColor

Sub **AddColor**(ByVal **Color** As **Color**)

Description

The **AddColor** method creates a new color table in Corel PHOTO-PAINT. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

Parameters	Description
Color	Specifies a color object that creates a new color table with the **AddColor** method. A color is an effect applied to an object that alters the object's appearance by the way it reflects light.

{button ,AL(`CLS_ColorTable;FNC_AddColor')} **Related Topics**

ColorTable.GetIndexOfColor

Function **GetIndexOfColor**(ByVal **Color** As **Color**) As Long

Description

The **GetIndexOfColor** method returns the index number of a color in the palette of a color table in Corel PHOTO-PAINT. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

Parameters	Description
Color	Specifies a color object and returns the index number of that color with the GetIndexOfColor method. A color is an effect applied to an object that alters the object's appearance by the way it reflects light.

{button ,AL(`CLS_ColorTable;FNC_GetIndexOfColor')} **Related Topics**

ColorTable.LoadPalette

Sub **LoadPalette**(ByVal **Palette** As **Palette**)

Description

The **LoadPalette** method loads a color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Parameters	Description
Palette	Specifies the color palette to open with the **LoadPalette** method.

{button ,AL(`CLS_ColorTable;FNC_LoadPalette')} **Related Topics**

ColorTable.LoadPaletteFromFile

Sub **LoadPaletteFromFile**(ByVal **FileName** As String)

Description

The **LoadPaletteFromFile** method loads a color palette that is stored in a file outside of Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Parameters	Description
FileName	Specifies the computer location of a color palette that is loaded with the **LoadPaletteFromFile** method.

{button ,AL(`CLS_ColorTable;FNC_LoadPaletteFromFile')} **Related Topics**

cdrSortReverse	0
cdrSortHue	1
cdrSortBrightness	2
cdrSortSaturation	3
cdrSortRGB	4
cdrSortHSB	5
cdrSortName	6

ColorTable.Sort

Sub **Sort**(ByVal **Method** As **cdrPaletteSortMethod**)

Description

The **Sort** method sorts the colors in a color table in Corel PHOTO-PAINT. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

Parameters	Description
Method	Specifies how to sort the colors in a color table. This value returns cdrPaletteSortMethod.

{button ,AL(`CLS_ColorTable;FNC_Sort')} **Related Topics**

CorelScript methods

CorelScript Legend

- AdjustEffectInfo
- AlphaOrder
- BindToActiveDocument
- BitmapEffect
- BrushDabSettings
- BrushOrbitSettings
- BrushSettings
- BrushTextureSettings
- BrushTool
- ChannelNew
- ChannelProperties
- CloneTool
- ColorMaskApply
- ColorMaskColor
- ColorMaskCreateChannel
- ColorMaskCreateMask
- ColorMaskReset
- ColorReplace
- ColorReplacerTool
- ContinueDraw
- CreateBackground
- DuotoneHandle
- DuotoneInfo
- EditCheckpoint
- EditClear
- EditClearClipboard
- EditCopy
- EditCopyToFile
- EditCopyVisible
- EditCut
- EditCutIntoSelection
- EditCutMask
- EditCutSelection
- EditFill
- EditPasteDocument
- EditPasteFromFile
- EditPasteIntoSelection
- EditPasteObject
- EditPasteSelection
- EditRestoreCheckpoint
- EditUndo
- Effect3DRotate
- Effect3DStereoNoise
- EffectAdaptiveUnsharp
- EffectAddNoise
- EffectAdjustBlur
- EffectAdjustNoise
- EffectAdjustSharpness
- EffectAlchemy
- EffectBandPass
- EffectBitPlanes

EffectBrickWall
EffectBubbles
EffectBumpMap
EffectCanvas
EffectCharcoal
EffectCobblestone
EffectConteCrayon
EffectCraft
EffectCrayon
EffectCrystalize
EffectCubist
EffectCylinder
EffectDabble
EffectDiffuse
EffectDirectionalSharpen
EffectDirectionalSmooth
EffectDisplace
EffectDustScratch
EffectEdgeDetect
EffectElephantSkin
EffectEmboss
EffectEtching
EffectFabric
EffectFindEdges
EffectFrame
EffectFrameSettings
EffectGaussianBlur
EffectGlass
EffectGlassBlock
EffectHalftone
EffectHighPass
EffectImpressionist
EffectJaggyDespeckle
EffectKidsplay
EffectLensFlare
EffectLighting
EffectLightSource
EffectLocalHistogram
EffectLowPass
EffectMapToObject
EffectMaximum
EffectMedian
EffectMeshPoint
EffectMeshWarp
EffectMinimum
EffectMosaic
EffectMotionBlur
EffectOffset
EffectPageCurl
EffectPaletteKnife
EffectParticle
EffectPastels
EffectPenInk
EffectPerspective
EffectPinchPunch

EffectPixelate
EffectPlasterWall
EffectPlastic
EffectPlugin
EffectPointillist
EffectPsychedelic
EffectPuzzle
EffectRadialBlur
EffectReliefSculpture
EffectRemoveMoire
EffectRemoveNoise
EffectRipple
EffectScatter
EffectScraperboard
EffectScreenDoor
EffectSharpen
EffectShear
EffectShearTable
EffectSketchPad
EffectSmokedGlass
EffectSmooth
EffectSoften
EffectSolarize
EffectSphere
EffectStainedGlass
EffectStone
EffectSwirl
EffectTheBoss
EffectTile
EffectTool
EffectTraceContour
EffectUnderpainting
EffectUnsharpMask
EffectUserDefined
EffectUserDefinedPoint
EffectVignette
EffectVortex
EffectWatercolor
EffectWaterMarker
EffectWavePaper
EffectWeather
EffectWetPaint
EffectWhirlpool
EffectWind
EffectZigZag
EffectZoom
Ellipse
EllipseTool
EndAdjustEffect
EndColorEffect
EndColorMask
EndColorReplace
EndColorTable
EndConvertDuotone
EndConvertPaletted

EndDraw
EndEditFill
EndEffectFrame
EndEffectLighting
EndEffectMeshWarp
EndEffectShear
EndEffectUserDefined
EndGuideline
EndImageEqualize
EndImageSTBalance
EndImageToneCurve
EndMaskCreate
EndMovieFrameRate
EndObject
EndObjectColorTransparencyTool
EndObjectTagWWWURL
Eraser
FadeLastCommand
FileClose
FileExit
FileImport
FileNew
FileOpen
FilePrint
FileRevert
FileSave
FileSelectPartialArea
Fill
FillBitmap
FillFountain
FillFountainApply
FillFountainColor
FillSolid
FillTexture
FillTextureSettings
FillTool
FilterAVI
FilterFPX
FilterGIF
FilterICO
FilterJPG
FilterMPG
FilterOS2
FilterPCD
FilterPNG
FilterRAW
FilterTGA
FilterWVL
GetChannelCount
GetChannelName
GetCurrentMovieFrame
GetDocumentCount
GetDocumentHeight
GetDocumentHRes
GetDocumentIsMovie

GetDocumentIsPartial
GetDocumentModified
GetDocumentName
GetDocumentType
GetDocumentVRes
GetDocumentWidth
GetDocumentXdpi
GetDocumentXGridFrequency
GetDocumentXRulerUnits
GetDocumentYdpi
GetDocumentYGridFrequency
GetDocumentYRulerUnits
GetFillColor
GetFrameCount
GetGuidelineCount
GetGuidelineProperties
GetMaskPresent
GetMaskRectangle
GetObjectCount
GetObjectIsEditable
GetObjectIsSelected
GetObjectIsVisible
GetObjectMergeMode
GetObjectName
GetObjectOpacity
GetObjectProperties
GetObjectRectangle
GetPaintColor
GetPaintVersion
GetPaperColor
GetPartialDocumentHeight
GetPartialDocumentWidth
GetPhotoPaintDir
GetPixelColor
GetSelectedObjectsRectangle
Gradient
GradientPoint
GradientTool
GridSetup
GuidelineAdd
GuidelineDelete
GuidelineMove
GuidelineSelect
ImageApplyICCProfile
ImageAutoEqualize
ImageBCI
ImageColorBalance
ImageColorCrop
ImageColorHue
ImageColorTable
ImageColorTone
ImageConvert
ImageConvertDuotone
ImageConvertPaletted
ImageConvertVideoNTSC

ImageConvertVideoPAL
ImageCrop
ImageCropToMask
ImageDeInterlace
ImageDesaturate
ImageDeskew
ImageDeskewCrop
ImageDuplicate
ImageEqualize
ImageEqualizeChannel
ImageFlipHorizontal
ImageFlipVertical
ImageGamma
ImageHSL
ImageHSLChannel
ImageInvert
ImageLevelThreshold
ImagePapersize
ImagePosterize
ImageReplaceColors
ImageResample
ImageRotate
ImageSelectiveColor
ImageSelectiveColorChannel
ImageSetChannel
ImageSplit
ImageSprayerList
ImageSprayerSettings
ImageSprayerTool
ImageSTBalance
ImageSTColor
ImageToneChannel
ImageToneCurve
ImageTonePoint
ImageToneTable
LensCreateFromMask
LensEdit
LensNew
LineTool
LocalUndo
MaskAffineDistort
MaskAlign
MaskAntiAlias
MaskAverage
MaskBorder
MaskBrush
MaskChannelAdd
MaskChannelDelete
MaskChannelLoad
MaskChannelName
MaskChannelSave
MaskChannelToMask
MaskCreate
MaskCreateFromPath
MaskDeFloat

MaskDefloatIntoSelection
MaskDistort
MaskEllipse
MaskExpand
MaskFeather
MaskFlipHorizontal
MaskFlipVertical
MaskFloaterMoveTo
MaskFloaterTranslate
MaskFreehand
MaskGrow
MaskInvert
MaskLasso
MaskLoad
MaskMagicWand
MaskPaint
MaskRectangle
MaskReduce
MaskRemove
MaskRemoveHoles
MaskRotate
MaskSave
MaskScissors
MaskSelectAll
MaskSimilar
MaskSkew
MaskSmooth
MaskStretch
MaskStroke
MaskThreshold
MaskToMaskChannel
MaskTranslate
MovieBackOne
MovieCreate
MovieDeleteFrame
MovieForward
MovieForwardOne
MovieFrameDelay
MovieFrameRate
MovieGotoFrame
MovieInsertFile
MovieInsertFrame
MovieMoveFrame
MovieRewind
MovieSelectPartial
NibSettings
ObjectAddClipMask
ObjectAddClipMaskFromMask
ObjectAddClipMaskFromTransparency
ObjectAffineDistort
ObjectAlign
ObjectClip
ObjectClipToParent
ObjectColorTransparencyTool
ObjectCombine

ObjectCreate
ObjectCreateFromBackground
ObjectDefringe
ObjectDelete
ObjectDistort
ObjectDropShadow
ObjectDropShadowCombine
ObjectDropShadowDelete
ObjectDropShadowSplit
ObjectDuplicate
ObjectEdit
ObjectEditAll
ObjectEditNone
ObjectEditSelected
ObjectEditTransparency
ObjectFeather
ObjectFlipHorizontal
ObjectFlipVertical
ObjectGroup
ObjectMerge
ObjectMergeMode
ObjectName
ObjectNew
ObjectOpacity
ObjectOrder
ObjectOrderChange
ObjectProperties
ObjectRemoveClipMask
ObjectRemoveMatte
ObjectRotate
ObjectSelect
ObjectSelectAll
ObjectSelection
ObjectSelectNone
ObjectSkew
ObjectStretch
ObjectTagWWWURL
ObjectThreshold
ObjectToggleClipMask
ObjectToggleLinkClipMask
ObjectTranslate
ObjectTransparencyTool
ObjectUngroup
ObjectURLInfo
ObjectVisible
OverprintColor
PaletteColor
PathCreate
PathCreateFromMask
PathDelete
PathDuplicate
PathEnd
PathImportVector
PathLoad
PathNew

PathNode
PathSave
PathSetClippingPath
PathStroke
PathVisible
PolygonTool
PressureSettings
RandomSeed
Rectangle
RectangleTool
RegisterObject
RepeatSettings
SelectionMoveTo
SelectionPlugin
SetActiveTool
SetDocumentInfo
SetDocVisible
SetPaintColor
SetPaperColor
SetVisible
SnapToGrid
StartCloneDraw
StartDraw
SymmetrySettings
TextAppend
TextEdit
TextFitToPath
TextRender
TextSetting
TextTool
ToleranceSettings
TransparencyBrushTool
UnRegisterObject

CorelScript

Class **CorelScript**

Methods Referenced By

{button ,AL(`CLS_CorelScript')} **Related Topics**

CorelScript.AdjustEffectInfo

Sub **AdjustEffectInfo**(ByVal **EffectID** As Long, ByVal **Value1** As Long, ByVal **Value2** As Long, ByVal **Value3** As Long)

Member of **CorelScript**

Parameters	Description
EffectID	Description of EffectID goes here
Value1	Description of Value1 goes here
Value2	Description of Value2 goes here
Value3	Description of Value3 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_AdjustEffectInfo')} **Related Topics**

CorelScript.AlphaOrder

Sub **AlphaOrder**(ByVal **CurrentIndex** As Long, ByVal **NewIndex** As Long)

Member of **CorelScript**

Parameters	Description
CurrentIndex	Description of CurrentIndex goes here
NewIndex	Description of NewIndex goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_AlphaOrder')} **Related Topics**

CorelScript.BindToActiveDocument

Sub **BindToActiveDocument**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BindToActiveDocument')} **Related Topics**

CorelScript.BitmapEffect

Sub **BitmapEffect**(ByVal **Name** As String, ByVal **Parameters** As String)

Member of **CorelScript**

Parameters	Description
Name	Description of Name goes here
Parameters	Description of Parameters goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BitmapEffect')} **Related Topics**

CorelScript.BrushDabSettings

Sub **BrushDabSettings**(ByVal **Dabs** As Long, ByVal **Spacing** As Long, ByVal **Spread** As Long, ByVal **FadeOut** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Luminance** As Long)

Member of **CorelScript**

Parameters	Description
Dabs	Description of Dabs goes here
Spacing	Description of Spacing goes here
Spread	Description of Spread goes here
FadeOut	Description of FadeOut goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Luminance	Description of Luminance goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BrushDabSettings')} **Related Topics**

CorelScript.BrushOrbitSettings

Sub **BrushOrbitSettings**(ByVal **Points** As Long, ByVal **Radius** As Long, ByVal **Rotation** As Long, ByVal **GrowRate** As Long, ByVal **GrowScale** As Long, ByVal **Center** As Boolean, ByVal **Clockwise** As Boolean, ByVal **ColorHue** As Long, ByVal **ColorSaturation** As Long, ByVal **ColorLightness** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Lightness** As Long)

Member of **CorelScript**

Parameters	Description
Points	Description of Points goes here
Radius	Description of Radius goes here
Rotation	Description of Rotation goes here
GrowRate	Description of GrowRate goes here
GrowScale	Description of GrowScale goes here
Center	Description of Center goes here
Clockwise	Description of Clockwise goes here
ColorHue	Description of ColorHue goes here
ColorSaturation	Description of ColorSaturation goes here
ColorLightness	Description of ColorLightness goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Lightness	Description of Lightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BrushOrbitSettings')} **Related Topics**

CorelScript.BrushSettings

Sub **BrushSettings**(ByVal **ToolID** As Long, ByVal **BrushID** As Long, ByVal **TypeID** As Long, ByVal **MergeMode** As Long, ByVal **Amount** As Long, ByVal **NibShape** As Long, ByVal **Size** As Long, ByVal **Transparency** As Long, ByVal **Rotate** As Long, ByVal **Flatten** As Long, ByVal **SoftEdge** As Long)

Member of **CorelScript**

Parameters	Description
ToolID	Description of ToolID goes here
BrushID	Description of BrushID goes here
TypeID	Description of TypeID goes here
MergeMode	Description of MergeMode goes here
Amount	Description of Amount goes here
NibShape	Description of NibShape goes here
Size	Description of Size goes here
Transparency	Description of Transparency goes here
Rotate	Description of Rotate goes here
Flatten	Description of Flatten goes here
SoftEdge	Description of SoftEdge goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BrushSettings')} **Related Topics**

CorelScript.BrushTextureSettings

Sub **BrushTextureSettings**(ByVal **TextureFile** As String, ByVal **BrushTexture** As Long, ByVal **EdgeTexture** As Long, ByVal **Bleed** As Long, ByVal **SustainColor** As Long, ByVal **Smoothing** As Long, ByVal **AntiAlias** As Boolean, ByVal **Cumulative** As Boolean, ByVal **Range** As Long, ByVal **MergedSource** As Boolean)

Member of **CorelScript**

Parameters	Description
TextureFile	Description of TextureFile goes here
BrushTexture	Description of BrushTexture goes here
EdgeTexture	Description of EdgeTexture goes here
Bleed	Description of Bleed goes here
SustainColor	Description of SustainColor goes here
Smoothing	Description of Smoothing goes here
AntiAlias	Description of AntiAlias goes here
Cumulative	Description of Cumulative goes here
Range	Description of Range goes here
MergedSource	Description of MergedSource goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BrushTextureSettings')} **Related Topics**

CorelScript.BrushTool

Sub **BrushTool**(ByVal **BrushID** As Long, ByVal **TypeID** As Long, ByVal **MergeMode** As Long, ByVal **Amount** As Long, ByVal **NibShape** As Long, ByVal **Size** As Long, ByVal **Transparency** As Long, ByVal **Rotate** As Long, ByVal **Flatten** As Long, ByVal **SoftEdge** As Long)

Member of **CorelScript**

Parameters	Description
BrushID	Description of BrushID goes here
TypeID	Description of TypeID goes here
MergeMode	Description of MergeMode goes here
Amount	Description of Amount goes here
NibShape	Description of NibShape goes here
Size	Description of Size goes here
Transparency	Description of Transparency goes here
Rotate	Description of Rotate goes here
Flatten	Description of Flatten goes here
SoftEdge	Description of SoftEdge goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_BrushTool')} **Related Topics**

CorelScript.ChannelNew

Sub **ChannelNew**(ByVal **Name** As String, ByVal **Opacity** As Long, ByVal **Invert** As Long, ByVal **FillWhite** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Name	Description of Name goes here
Opacity	Description of Opacity goes here
Invert	Description of Invert goes here
FillWhite	Description of FillWhite goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ChannelNew')} **Related Topics**

CorelScript.ChannelProperties

Sub **ChannelProperties**(ByVal **ChannelID** As Long, ByVal **Name** As String, ByVal **Opacity** As Long, ByVal **Invert** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ChannelID	Description of ChannelID goes here
Name	Description of Name goes here
Opacity	Description of Opacity goes here
Invert	Description of Invert goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ChannelProperties')} **Related Topics**

CorelScript.CloneTool

Sub **CloneTool**(ByVal **BrushID** As Long, ByVal **TypeID** As Long, ByVal **MergeMode** As Long, ByVal **Amount** As Long, ByVal **NibShape** As Long, ByVal **Size** As Long, ByVal **Transparency** As Long, ByVal **Rotate** As Long, ByVal **Flatten** As Long, ByVal **SoftEdge** As Long)

Member of **CorelScript**

Parameters	Description
BrushID	Description of BrushID goes here
TypeID	Description of TypeID goes here
MergeMode	Description of MergeMode goes here
Amount	Description of Amount goes here
NibShape	Description of NibShape goes here
Size	Description of Size goes here
Transparency	Description of Transparency goes here
Rotate	Description of Rotate goes here
Flatten	Description of Flatten goes here
SoftEdge	Description of SoftEdge goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_CloneTool')} **Related Topics**

CorelScript.ColorMaskApply

Sub **ColorMaskApply**(ByVal **MaskMode** As Long, ByVal **Smoothing** As Long, ByVal **ToleranceMode** As Long)

Member of **CorelScript**

Parameters	Description
MaskMode	Description of MaskMode goes here
Smoothing	Description of Smoothing goes here
ToleranceMode	Description of ToleranceMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorMaskApply')} **Related Topics**

CorelScript.ColorMaskColor

Sub **ColorMaskColor**(ByVal **Number** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Number	Description of Number goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorMaskColor')} **Related Topics**

CorelScript.ColorMaskCreateChannel

Sub **ColorMaskCreateChannel**(ByVal **MaskMode** As Long, ByVal **Smoothing** As Long, ByVal **ToleranceMode** As Long, ByVal **Gamut** As Boolean, ByVal **Threshold** As Long, ByVal **BrightnessModel** As Long, ByVal **Name** As String)

Member of **CorelScript**

Parameters	Description
MaskMode	Description of MaskMode goes here
Smoothing	Description of Smoothing goes here
ToleranceMode	Description of ToleranceMode goes here
Gamut	Description of Gamut goes here
Threshold	Description of Threshold goes here
BrightnessModel	Description of BrightnessModel goes here
Name	Description of Name goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorMaskCreateChannel')} **Related Topics**

CorelScript.ColorMaskCreateMask

Sub **ColorMaskCreateMask**(ByVal **DrawMode** As Long, ByVal **MaskMode** As Long, ByVal **Smoothing** As Long, ByVal **ToleranceMode** As Long, ByVal **Gamut** As Boolean, ByVal **Threshold** As Long, ByVal **BrightnessModel** As Long)

Member of **CorelScript**

Parameters	Description
DrawMode	Description of DrawMode goes here
MaskMode	Description of MaskMode goes here
Smoothing	Description of Smoothing goes here
ToleranceMode	Description of ToleranceMode goes here
Gamut	Description of Gamut goes here
Threshold	Description of Threshold goes here
BrightnessModel	Description of BrightnessModel goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorMaskCreateMask')} **Related Topics**

CorelScript.ColorMaskReset

Sub **ColorMaskReset**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorMaskReset')} **Related Topics**

CorelScript.ColorReplace

Sub **ColorReplace**(ByVal **ToleranceMode** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
ToleranceMode	Description of ToleranceMode goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorReplace')} **Related Topics**

CorelScript.ColorReplacerTool

Sub **ColorReplacerTool**(ByVal **Width** As Long, ByVal **Flatten** As Long, ByVal **Rotate** As Long, ByVal **NibShape** As Long, ByVal **Transparency** As Long, ByVal **SoftEdge** As Long, ByVal **AntiAlias** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Flatten	Description of Flatten goes here
Rotate	Description of Rotate goes here
NibShape	Description of NibShape goes here
Transparency	Description of Transparency goes here
SoftEdge	Description of SoftEdge goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ColorReplacerTool')} **Related Topics**

CorelScript.ContinueDraw

Sub **ContinueDraw**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **Timer** As Long, ByVal **Pressure** As Long, ByVal **Tilt** As Long, ByVal **Rotate** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
Timer	Description of Timer goes here
Pressure	Description of Pressure goes here
Tilt	Description of Tilt goes here
Rotate	Description of Rotate goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ContinueDraw')} **Related Topics**

CorelScript.CreateBackground

Sub **CreateBackground**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_CreateBackground')} **Related Topics**

CorelScript.DuotoneHandle

Sub **DuotoneHandle**(ByVal **ToneNumber** As Long, ByVal **HandleNumber** As Long, ByVal **X** As Long, ByVal **Y** As Long)

Member of **CorelScript**

Parameters	Description
ToneNumber	Description of ToneNumber goes here
HandleNumber	Description of HandleNumber goes here
X	Description of X goes here
Y	Description of Y goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_DuotoneHandle')} **Related Topics**

CorelScript.DuotoneInfo

Sub **DuotoneInfo**(ByVal **ToneNumber** As Long, ByVal **Handles** As Long, ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long, ByVal **Black** As Long)

Member of **CorelScript**

Parameters	Description
ToneNumber	Description of ToneNumber goes here
Handles	Description of Handles goes here
Cyan	Description of Cyan goes here
Magenta	Description of Magenta goes here
Yellow	Description of Yellow goes here
Black	Description of Black goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_DuotoneInfo')} **Related Topics**

CorelScript.EditCheckpoint

Sub **EditCheckpoint**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCheckpoint')} **Related Topics**

CorelScript.EditClear

Sub **EditClear**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditClear')} **Related Topics**

CorelScript.EditClearClipboard

Sub **EditClearClipboard**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditClearClipboard')} **Related Topics**

CorelScript.EditCopy

Sub **EditCopy**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCopy')} **Related Topics**

CorelScript.EditCopyToFile

Sub **EditCopyToFile**(ByVal **FileName** As String, ByVal **FilterID** As Long, ByVal **Compression** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
FilterID	Description of FilterID goes here
Compression	Description of Compression goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCopyToFile')} **Related Topics**

CorelScript.EditCopyVisible

Sub **EditCopyVisible**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCopyVisible')} **Related Topics**

CorelScript.EditCut

Sub **EditCut**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCut')} **Related Topics**

CorelScript.EditCutIntoSelection

Sub **EditCutIntoSelection**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCutIntoSelection')} **Related Topics**

CorelScript.EditCutMask

Sub **EditCutMask**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCutMask')} **Related Topics**

CorelScript.EditCutSelection

Sub **EditCutSelection**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditCutSelection')} **Related Topics**

CorelScript.EditFill

Sub **EditFill**(ByVal **MergeMode** As Long, ByVal **StartTransparency** As Long, ByVal **EndTransparency** As Long, ByVal **GradientType** As Long, ByVal **Handles** As Long, ByVal **x1** As Long, ByVal **y1** As Long, ByVal **x2** As Long, ByVal **y2** As Long, ByVal **x3** As Long, ByVal **y3** As Long)

Member of **CorelScript**

Parameters	Description
MergeMode	Description of MergeMode goes here
StartTransparency	Description of StartTransparency goes here
EndTransparency	Description of EndTransparency goes here
GradientType	Description of GradientType goes here
Handles	Description of Handles goes here
x1	Description of x1 goes here
y1	Description of y1 goes here
x2	Description of x2 goes here
y2	Description of y2 goes here
x3	Description of x3 goes here
y3	Description of y3 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditFill')} **Related Topics**

CorelScript.EditPasteDocument

Sub **EditPasteDocument**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditPasteDocument')} **Related Topics**

CorelScript.EditPasteFromFile

Sub **EditPasteFromFile**(ByVal **FileName** As String, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **LoadType** As Long, ByVal **X** As Long, ByVal **Y** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
LoadType	Description of LoadType goes here
X	Description of X goes here
Y	Description of Y goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditPasteFromFile')} **Related Topics**

CorelScript.EditPasteIntoSelection

Sub **EditPasteIntoSelection**(ByVal **FileName** As String)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditPasteIntoSelection')} **Related Topics**

CorelScript.EditPasteObject

Sub **EditPasteObject**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **FileName** As String)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
FileName	Description of FileName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditPasteObject')} **Related Topics**

CorelScript.EditPasteSelection

Sub **EditPasteSelection**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **FileName** As String)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
FileName	Description of FileName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditPasteSelection')} **Related Topics**

CorelScript.EditRestoreCheckpoint

Sub **EditRestoreCheckpoint**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditRestoreCheckpoint')} **Related Topics**

CorelScript.EditUndo

Sub **EditUndo**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EditUndo')} **Related Topics**

CorelScript.Effect3DRotate

Sub **Effect3DRotate**(ByVal **Horizontal** As Long, ByVal **Vertical** As Long, ByVal **Face** As Long, ByVal **BestFit** As Boolean)

Member of **CorelScript**

Parameters	Description
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here
Face	Description of Face goes here
BestFit	Description of BestFit goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Effect3DRotate')} **Related Topics**

CorelScript.Effect3DStereoNoise

Sub **Effect3DStereoNoise**(ByVal **Depth** As Long, ByVal **Dots** As Boolean)

Member of **CorelScript**

Parameters	Description
Depth	Description of Depth goes here
Dots	Description of Dots goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Effect3DStereoNoise')} **Related Topics**

CorelScript.EffectAdaptiveUnsharp

Sub **EffectAdaptiveUnsharp**(ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectAdaptiveUnsharp')} **Related Topics**

CorelScript.EffectAddNoise

Sub **EffectAddNoise**(ByVal **Level** As Long, ByVal **Density** As Long, ByVal **ColorNoise** As Boolean, ByVal **NoiseType** As Long, ByVal **ColorMode** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here
Density	Description of Density goes here
ColorNoise	Description of ColorNoise goes here
NoiseType	Description of NoiseType goes here
ColorMode	Description of ColorMode goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectAddNoise')} **Related Topics**

CorelScript.EffectAdjustBlur

Sub **EffectAdjustBlur**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectAdjustBlur')} **Related Topics**

CorelScript.EffectAdjustNoise

Sub **EffectAdjustNoise**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectAdjustNoise')} **Related Topics**

CorelScript.EffectAdjustSharpness

Sub **EffectAdjustSharpness**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectAdjustSharpness')} **Related Topics**

CorelScript.EffectAlchemy

Sub **EffectAlchemy**(ByVal **FileName** As String, ByVal **Layer** As Long, ByVal **Density** As Long, ByVal **Rand** As Long, ByVal **HPosV** As Long, ByVal **VPosV** As Long, ByVal **BrushClr** As Boolean, ByVal **BackClr** As Boolean, ByVal **ClrModel** As Long, ByVal **ClrP1** As Long, ByVal **ClrP2** As Long, ByVal **ClrP3** As Long, ByVal **ClrP4** As Long, ByVal **ClrMode2** As Long, ByVal **ClrP5** As Long, ByVal **ClrP6** As Long, ByVal **ClrP7** As Long, ByVal **ClrP8** As Long, ByVal **HueV** As Long, ByVal **SatV** As Long, ByVal **BrtV** As Long, ByVal **BrushSize** As Long, ByVal **Size1** As Long, ByVal **Size2** As Long, ByVal **SizeV** As Long, ByVal **BrushAngle** As Long, ByVal **Angle1** As Long, ByVal **Angle2** As Long, ByVal **AngleV** As Long, ByVal **BrushTrans** As Long, ByVal **Trans1** As Long, ByVal **Trans2** As Long, ByVal **TransV** As Long, ByVal **Dx** As Double, ByVal **Dy** As Double)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Layer	Description of Layer goes here
Density	Description of Density goes here
Rand	Description of Rand goes here
HPosV	Description of HPosV goes here
VPosV	Description of VPosV goes here
BrushClr	Description of BrushClr goes here
BackClr	Description of BackClr goes here
ClrModel	Description of ClrModel goes here
ClrP1	Description of ClrP1 goes here
ClrP2	Description of ClrP2 goes here
ClrP3	Description of ClrP3 goes here
ClrP4	Description of ClrP4 goes here
ClrMode2	Description of ClrMode2 goes here
ClrP5	Description of ClrP5 goes here
ClrP6	Description of ClrP6 goes here
ClrP7	Description of ClrP7 goes here
ClrP8	Description of ClrP8 goes here
HueV	Description of HueV goes here
SatV	Description of SatV goes here
BrtV	Description of BrtV goes here
BrushSize	Description of BrushSize goes here
Size1	Description of Size1 goes here
Size2	Description of Size2 goes here
SizeV	Description of SizeV goes here
BrushAngle	Description of BrushAngle goes here
Angle1	Description of Angle1 goes here
Angle2	Description of Angle2 goes here
AngleV	Description of AngleV goes here
BrushTrans	Description of BrushTrans goes here
Trans1	Description of Trans1 goes here
Trans2	Description of Trans2 goes here

TransV	Description of TransV goes here
Dx	Description of Dx goes here
Dy	Description of Dy goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectAlchemy')} **Related Topics**

CorelScript.EffectBandPass

Sub **EffectBandPass**(ByVal **InRadius** As Long, ByVal **OutRadius** As Long, ByVal **InBand** As Long, ByVal **MidBand** As Long, ByVal **OutBand** As Long)

Member of **CorelScript**

Parameters	Description
InRadius	Description of InRadius goes here
OutRadius	Description of OutRadius goes here
InBand	Description of InBand goes here
MidBand	Description of MidBand goes here
OutBand	Description of OutBand goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectBandPass')} **Related Topics**

CorelScript.EffectBitPlanes

Sub **EffectBitPlanes**(ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long)

Member of **CorelScript**

Parameters	Description
Red	Description of Red goes here
Green	Description of Green goes here
Blue	Description of Blue goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectBitPlanes')} **Related Topics**

CorelScript.EffectBrickWall

Sub **EffectBrickWall**(ByVal **Roughness** As Long, ByVal **BrickWidth** As Long, ByVal **BrickHeight** As Long, ByVal **GroutWidth** As Long, ByVal **Direction** As Long)

Member of **CorelScript**

Parameters	Description
Roughness	Description of Roughness goes here
BrickWidth	Description of BrickWidth goes here
BrickHeight	Description of BrickHeight goes here
GroutWidth	Description of GroutWidth goes here
Direction	Description of Direction goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectBrickWall')} **Related Topics**

CorelScript.EffectBubbles

Sub **EffectBubbles**(ByVal **Diameter** As Long, ByVal **Coverage** As Long, ByVal **Direction** As Long, ByVal **Refraction** As Long, ByVal **Seed** As Long)

Member of **CorelScript**

Parameters	Description
Diameter	Description of Diameter goes here
Coverage	Description of Coverage goes here
Direction	Description of Direction goes here
Refraction	Description of Refraction goes here
Seed	Description of Seed goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectBubbles')} **Related Topics**

CorelScript.EffectBumpMap

Sub **EffectBumpMap**(ByVal **FileName** As String, ByVal **Mode** As Long, ByVal **TileWidth** As Long, ByVal **TileHeight** As Long, ByVal **Floor** As Long, ByVal **Ceiling** As Long, ByVal **ScaleFactor** As Long, ByVal **Highlight** As Long, ByVal **Invert** As Boolean, ByVal **Smooth** As Boolean, ByVal **AmbientBrightness** As Long, ByVal **AmbientColorModel** As Long, ByVal **AmbientColor1** As Long, ByVal **AmbientColor2** As Long, ByVal **AmbientColor3** As Long, ByVal **AmbientColor4** As Long, ByVal **DirectionalBrightness** As Long, ByVal **DirectionalColorModel** As Long, ByVal **DirectionalColor1** As Long, ByVal **DirectionalColor2** As Long, ByVal **DirectionalColor3** As Long, ByVal **DirectionalColor4** As Long, ByVal **Direction** As Long, ByVal **Declination** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Mode	Description of Mode goes here
TileWidth	Description of TileWidth goes here
TileHeight	Description of TileHeight goes here
Floor	Description of Floor goes here
Ceiling	Description of Ceiling goes here
ScaleFactor	Description of ScaleFactor goes here
Highlight	Description of Highlight goes here
Invert	Description of Invert goes here
Smooth	Description of Smooth goes here
AmbientBrightness	Description of AmbientBrightness goes here
AmbientColorModel	Description of AmbientColorModel goes here
AmbientColor1	Description of AmbientColor1 goes here
AmbientColor2	Description of AmbientColor2 goes here
AmbientColor3	Description of AmbientColor3 goes here
AmbientColor4	Description of AmbientColor4 goes here
DirectionalBrightness	Description of DirectionalBrightness goes here
DirectionalColorModel	Description of DirectionalColorModel goes here
DirectionalColor1	Description of DirectionalColor1 goes here
DirectionalColor2	Description of DirectionalColor2 goes here
DirectionalColor3	Description of DirectionalColor3 goes here
DirectionalColor4	Description of DirectionalColor4 goes here
Direction	Description of Direction goes here
Declination	Description of Declination goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectBumpMap')} **Related Topics**

CorelScript.EffectCanvas

Sub **EffectCanvas**(ByVal **FileName** As String, ByVal **Transparency** As Long, ByVal **Emboss** As Long, ByVal **X** As Long, ByVal **Y** As Long, ByVal **Mode** As Long, ByVal **Offset** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Transparency	Description of Transparency goes here
Emboss	Description of Emboss goes here
X	Description of X goes here
Y	Description of Y goes here
Mode	Description of Mode goes here
Offset	Description of Offset goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCanvas')} **Related Topics**

CorelScript.EffectCharcoal

Sub **EffectCharcoal**(ByVal **Size** As Long, ByVal **Edge** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Edge	Description of Edge goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCharcoal')} **Related Topics**

CorelScript.EffectCobblestone

Sub **EffectCobblestone**(ByVal **Roughness** As Long, ByVal **Size** As Long, ByVal **GroutWidth** As Long, ByVal **Direction** As Long, ByVal **Warp** As Boolean)

Member of **CorelScript**

Parameters	Description
Roughness	Description of Roughness goes here
Size	Description of Size goes here
GroutWidth	Description of GroutWidth goes here
Direction	Description of Direction goes here
Warp	Description of Warp goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCobblestone')} **Related Topics**

CorelScript.EffectConteCrayon

Sub **EffectConteCrayon**(ByVal **ConteColor** As Long, ByVal **Pressure** As Long, ByVal **Texture** As Long, ByVal **Type** As Long, ByVal **Comp1** As Long, ByVal **Comp2** As Long, ByVal **Comp3** As Long, ByVal **Comp4** As Long)

Member of **CorelScript**

Parameters	Description
ConteColor	Description of ConteColor goes here
Pressure	Description of Pressure goes here
Texture	Description of Texture goes here
Type	Description of Type goes here
Comp1	Description of Comp1 goes here
Comp2	Description of Comp2 goes here
Comp3	Description of Comp3 goes here
Comp4	Description of Comp4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectConteCrayon')} **Related Topics**

CorelScript.EffectCraft

Sub **EffectCraft**(ByVal **Style** As Long, ByVal **Complete** As Long, ByVal **Brightness** As Long, ByVal **Size** As Long, ByVal **Rotation** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Complete	Description of Complete goes here
Brightness	Description of Brightness goes here
Size	Description of Size goes here
Rotation	Description of Rotation goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCraft')} **Related Topics**

CorelScript.EffectCrayon

Sub **EffectCrayon**(ByVal **Size** As Long, ByVal **Outline** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Outline	Description of Outline goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCrayon')} **Related Topics**

CorelScript.EffectCrystalize

Sub **EffectCrystalize**(ByVal **Size** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCrystalize')} **Related Topics**

CorelScript.EffectCubist

Sub **EffectCubist**(ByVal **Size** As Long, ByVal **Brightness** As Long, ByVal **Type** As Long, ByVal **Comp1** As Long, ByVal **Comp2** As Long, ByVal **Comp3** As Long, ByVal **Comp4** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Brightness	Description of Brightness goes here
Type	Description of Type goes here
Comp1	Description of Comp1 goes here
Comp2	Description of Comp2 goes here
Comp3	Description of Comp3 goes here
Comp4	Description of Comp4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCubist')} **Related Topics**

CorelScript.EffectCylinder

Sub **EffectCylinder**(ByVal **Mode** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectCylinder')} **Related Topics**

CorelScript.EffectDabble

Sub **EffectDabble**(ByVal **Distribution** As Long, ByVal **Style** As Long, ByVal **Size** As Long)

Member of **CorelScript**

Parameters	Description
Distribution	Description of Distribution goes here
Style	Description of Style goes here
Size	Description of Size goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectDabble')} **Related Topics**

CorelScript.EffectDiffuse

Sub **EffectDiffuse**(ByVal **Level** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectDiffuse')} **Related Topics**

CorelScript.EffectDirectionalSharpen

Sub **EffectDirectionalSharpen**(ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	**Description**
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectDirectionalSharpen')} **Related Topics**

CorelScript.EffectDirectionalSmooth

Sub **EffectDirectionalSmooth**(ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectDirectionalSmooth')} **Related Topics**

CorelScript.EffectDisplace

Sub **EffectDisplace**(ByVal **FileName** As String, ByVal **Displacement** As Long, ByVal **Edges** As Long, ByVal **Horizontal** As Long, ByVal **Vertical** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Displacement	Description of Displacement goes here
Edges	Description of Edges goes here
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectDisplace')} **Related Topics**

CorelScript.EffectDustScratch

Sub **EffectDustScratch**(ByVal **Level** As Long, ByVal **Radius** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here
Radius	Description of Radius goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectDustScratch')} **Related Topics**

CorelScript.EffectEdgeDetect

Sub **EffectEdgeDetect**(ByVal **Sensitivity** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Sensitivity	Description of Sensitivity goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectEdgeDetect')} **Related Topics**

CorelScript.EffectElephantSkin

Sub **EffectElephantSkin**(ByVal **Age** As Long, ByVal **Variation** As Long, ByVal **Type** As Long, ByVal **Comp1** As Long, ByVal **Comp2** As Long, ByVal **Comp3** As Long, ByVal **Comp4** As Long)

Member of **CorelScript**

Parameters	Description
Age	Description of Age goes here
Variation	Description of Variation goes here
Type	Description of Type goes here
Comp1	Description of Comp1 goes here
Comp2	Description of Comp2 goes here
Comp3	Description of Comp3 goes here
Comp4	Description of Comp4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectElephantSkin')} **Related Topics**

CorelScript.EffectEmboss

Sub **EffectEmboss**(ByVal **Depth** As Long, ByVal **Level** As Long, ByVal **Direction** As Long, ByVal **Color** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Depth	Description of Depth goes here
Level	Description of Level goes here
Direction	Description of Direction goes here
Color	Description of Color goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectEmboss')} **Related Topics**

CorelScript.EffectEtching

Sub **EffectEtching**(ByVal **Detail** As Long, ByVal **Depth** As Long, ByVal **Direction** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Detail	Description of Detail goes here
Depth	Description of Depth goes here
Direction	Description of Direction goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectEtching')} **Related Topics**

CorelScript.EffectFabric

Sub **EffectFabric**(ByVal **Style** As Long, ByVal **Complete** As Long, ByVal **Brightness** As Long, ByVal **Size** As Long, ByVal **Rotation** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Complete	Description of Complete goes here
Brightness	Description of Brightness goes here
Size	Description of Size goes here
Rotation	Description of Rotation goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectFabric')} **Related Topics**

CorelScript.EffectFindEdges

Sub **EffectFindEdges**(ByVal **Level** As Long, ByVal **EdgeType** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here
EdgeType	Description of EdgeType goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectFindEdges')} **Related Topics**

CorelScript.EffectFrame

Sub **EffectFrame**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectFrame')} **Related Topics**

CorelScript.EffectFrameSettings

Sub **EffectFrameSettings**(ByVal **Number** As Long, ByVal **FileName** As String, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **PosX** As Double, ByVal **PosY** As Double, ByVal **Opacity** As Long, ByVal **Fade** As Long, ByVal **Horizontal** As Long, ByVal **Vertical** As Long, ByVal **Angle** As Long, ByVal **HFlip** As Long, ByVal **VFlip** As Long, ByVal **BlendMode** As Long, ByVal **UseImage** As Long, ByVal **UsePath** As Long)

Member of **CorelScript**

Parameters	Description
Number	Description of Number goes here
FileName	Description of FileName goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
PosX	Description of PosX goes here
PosY	Description of PosY goes here
Opacity	Description of Opacity goes here
Fade	Description of Fade goes here
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here
Angle	Description of Angle goes here
HFlip	Description of HFlip goes here
VFlip	Description of VFlip goes here
BlendMode	Description of BlendMode goes here
UseImage	Description of UseImage goes here
UsePath	Description of UsePath goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectFrameSettings')} **Related Topics**

CorelScript.EffectGaussianBlur

Sub **EffectGaussianBlur**(ByVal **Radius** As Double)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectGaussianBlur')} **Related Topics**

CorelScript.EffectGlass

Sub **EffectGlass**(ByVal **Angle** As Long, ByVal **BevelWidth** As Long, ByVal **Brightness** As Long, ByVal **Direction** As Long, ByVal **Dropoff** As Long, ByVal **Opacity** As Long, ByVal **Refraction** As Long, ByVal **Sharpness** As Long, ByVal **Smooth** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Angle	Description of Angle goes here
BevelWidth	Description of BevelWidth goes here
Brightness	Description of Brightness goes here
Direction	Description of Direction goes here
Dropoff	Description of Dropoff goes here
Opacity	Description of Opacity goes here
Refraction	Description of Refraction goes here
Sharpness	Description of Sharpness goes here
Smooth	Description of Smooth goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectGlass')} **Related Topics**

CorelScript.EffectGlassBlock

Sub **EffectGlassBlock**(ByVal **Width** As Long, ByVal **Height** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectGlassBlock')} **Related Topics**

CorelScript.EffectHalftone

Sub **EffectHalftone**(ByVal **Radius** As Long, ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long, ByVal **Black** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Cyan	Description of Cyan goes here
Magenta	Description of Magenta goes here
Yellow	Description of Yellow goes here
Black	Description of Black goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectHalftone')} **Related Topics**

CorelScript.EffectHighPass

Sub **EffectHighPass**(ByVal **Radius** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectHighPass')} **Related Topics**

CorelScript.EffectImpressionist

Sub **EffectImpressionist**(ByVal **Style** As Long, ByVal **Detail** As Long, ByVal **Coloration** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Detail	Description of Detail goes here
Coloration	Description of Coloration goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectImpressionist')} **Related Topics**

CorelScript.EffectJaggyDespeckle

Sub **EffectJaggyDespeckle**(ByVal **Width** As Long, ByVal **Height** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectJaggyDespeckle')} **Related Topics**

CorelScript.EffectKidsplay

Sub **EffectKidsplay**(ByVal **Game** As Long, ByVal **Complete** As Long, ByVal **Brightness** As Long, ByVal **Size** As Long, ByVal **Rotation** As Long)

Member of **CorelScript**

Parameters	Description
Game	Description of Game goes here
Complete	Description of Complete goes here
Brightness	Description of Brightness goes here
Size	Description of Size goes here
Rotation	Description of Rotation goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectKidsplay')} **Related Topics**

CorelScript.EffectLensFlare

Sub **EffectLensFlare**(ByVal **X** As Double, ByVal **Y** As Double, ByVal **Brightness** As Long, ByVal **LensType** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
Brightness	Description of Brightness goes here
LensType	Description of LensType goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectLensFlare')} **Related Topics**

CorelScript.EffectLighting

Sub **EffectLighting**(ByVal **Sources** As Long)

Member of **CorelScript**

Parameters	Description
Sources	Description of Sources goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectLighting')} **Related Topics**

CorelScript.EffectLightSource

Sub **EffectLightSource**(ByVal **Index** As Long, ByVal **X** As Double, ByVal **Y** As Double, ByVal **Type** As Long, ByVal **Height** As Long, ByVal **Direction** As Long, ByVal **Elevation** As Long, ByVal **Intensity** As Long, ByVal **Aperture** As Long, ByVal **Focus** As Long, ByVal **Whiteness** As Long, ByVal **Exposure** As Long, ByVal **Channel** As Long, ByVal **Depth** As Long, ByVal **Contrast** As Long, ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long, ByVal **Switch** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
X	Description of X goes here
Y	Description of Y goes here
Type	Description of Type goes here
Height	Description of Height goes here
Direction	Description of Direction goes here
Elevation	Description of Elevation goes here
Intensity	Description of Intensity goes here
Aperture	Description of Aperture goes here
Focus	Description of Focus goes here
Whiteness	Description of Whiteness goes here
Exposure	Description of Exposure goes here
Channel	Description of Channel goes here
Depth	Description of Depth goes here
Contrast	Description of Contrast goes here
Red	Description of Red goes here
Green	Description of Green goes here
Blue	Description of Blue goes here
Switch	Description of Switch goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectLightSource')} **Related Topics**

CorelScript.EffectLocalHistogram

Sub **EffectLocalHistogram**(ByVal **Width** As Long, ByVal **Height** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectLocalHistogram')} **Related Topics**

CorelScript.EffectLowPass

Sub **EffectLowPass**(ByVal **Radius** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectLowPass')} **Related Topics**

CorelScript.EffectMapToObject

Sub **EffectMapToObject**(ByVal **Mode** As Long, ByVal **Percentage** As Long, ByVal **Quality** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here
Percentage	Description of Percentage goes here
Quality	Description of Quality goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMapToObject')} **Related Topics**

CorelScript.EffectMaximum

Sub **EffectMaximum**(ByVal **Radius** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMaximum')} **Related Topics**

CorelScript.EffectMedian

Sub **EffectMedian**(ByVal **Radius** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMedian')} **Related Topics**

CorelScript.EffectMeshPoint

Sub **EffectMeshPoint**(ByVal **Row** As Long, ByVal **Column** As Long, ByVal **X** As Double, ByVal **Y** As Double)

Member of **CorelScript**

Parameters	Description
Row	Description of Row goes here
Column	Description of Column goes here
X	Description of X goes here
Y	Description of Y goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMeshPoint')} **Related Topics**

CorelScript.EffectMeshWarp

Sub **EffectMeshWarp**(ByVal **Width** As Long, ByVal **Height** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMeshWarp')} **Related Topics**

CorelScript.EffectMinimum

Sub **EffectMinimum**(ByVal **Radius** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMinimum')} **Related Topics**

CorelScript.EffectMosaic

Sub **EffectMosaic**(ByVal **Size** As Long, ByVal **Vignette** As Long, ByVal **Type** As Long, ByVal **Comp1** As Long, ByVal **Comp2** As Long, ByVal **Comp3** As Long, ByVal **Comp4** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Vignette	Description of Vignette goes here
Type	Description of Type goes here
Comp1	Description of Comp1 goes here
Comp2	Description of Comp2 goes here
Comp3	Description of Comp3 goes here
Comp4	Description of Comp4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMosaic')} **Related Topics**

CorelScript.EffectMotionBlur

Sub **EffectMotionBlur**(ByVal **Speed** As Long, ByVal **Direction** As Long, ByVal **Method** As Long)

Member of **CorelScript**

Parameters	Description
Speed	Description of Speed goes here
Direction	Description of Direction goes here
Method	Description of Method goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectMotionBlur')} **Related Topics**

CorelScript.EffectOffset

Sub **EffectOffset**(ByVal **Horizontal** As Long, ByVal **Vertical** As Long, ByVal **Shift** As Boolean, ByVal **Edges** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here
Shift	Description of Shift goes here
Edges	Description of Edges goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectOffset')} **Related Topics**

CorelScript.EffectPageCurl

Sub **EffectPageCurl**(ByVal **Corner** As Long, ByVal **Direction** As Boolean, ByVal **Width** As Long, ByVal **Height** As Long, ByVal **Opaque** As Boolean, ByVal **CurlColorModel** As Long, ByVal **CurlColor1** As Long, ByVal **CurlColor2** As Long, ByVal **CurlColor3** As Long, ByVal **CurlColor4** As Long, ByVal **BackColorModel** As Long, ByVal **BackColor1** As Long, ByVal **BackColor2** As Long, ByVal **BackColor3** As Long, ByVal **BackColor4** As Long)

Member of **CorelScript**

Parameters	Description
Corner	Description of Corner goes here
Direction	Description of Direction goes here
Width	Description of Width goes here
Height	Description of Height goes here
Opaque	Description of Opaque goes here
CurlColorModel	Description of CurlColorModel goes here
CurlColor1	Description of CurlColor1 goes here
CurlColor2	Description of CurlColor2 goes here
CurlColor3	Description of CurlColor3 goes here
CurlColor4	Description of CurlColor4 goes here
BackColorModel	Description of BackColorModel goes here
BackColor1	Description of BackColor1 goes here
BackColor2	Description of BackColor2 goes here
BackColor3	Description of BackColor3 goes here
BackColor4	Description of BackColor4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPageCurl')} **Related Topics**

CorelScript.EffectPaletteKnife

Sub **EffectPaletteKnife**(ByVal **BladeSize** As Long, ByVal **SoftEdge** As Long, ByVal **Angle** As Long)

Member of **CorelScript**

Parameters	Description
BladeSize	Description of BladeSize goes here
SoftEdge	Description of SoftEdge goes here
Angle	Description of Angle goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPaletteKnife')} **Related Topics**

CorelScript.EffectParticle

Sub **EffectParticle**(ByVal **Style** As Long, ByVal **Size** As Long, ByVal **Density** As Long, ByVal **Coloration** As Long, ByVal **Transparency** As Long, ByVal **Angle** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Size	Description of Size goes here
Density	Description of Density goes here
Coloration	Description of Coloration goes here
Transparency	Description of Transparency goes here
Angle	Description of Angle goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectParticle')} **Related Topics**

CorelScript.EffectPastels

Sub **EffectPastels**(ByVal **Type** As Long, ByVal **Size** As Long, ByVal **Hue** As Long)

Member of **CorelScript**

Parameters	Description
Type	Description of Type goes here
Size	Description of Size goes here
Hue	Description of Hue goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPastels')} **Related Topics**

CorelScript.EffectPenInk

Sub **EffectPenInk**(ByVal **Style** As Long, ByVal **Density** As Long, ByVal **Inkpool** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Density	Description of Density goes here
Inkpool	Description of Inkpool goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPenInk')} **Related Topics**

CorelScript.EffectPerspective

Sub **EffectPerspective**(ByVal **x1** As Double, ByVal **y1** As Double, ByVal **x2** As Double, ByVal **y2** As Double, ByVal **x3** As Double, ByVal **y3** As Double, ByVal **x4** As Double, ByVal **y4** As Double, ByVal **BestFit** As Boolean)

Member of **CorelScript**

Parameters	Description
x1	Description of x1 goes here
y1	Description of y1 goes here
x2	Description of x2 goes here
y2	Description of y2 goes here
x3	Description of x3 goes here
y3	Description of y3 goes here
x4	Description of x4 goes here
y4	Description of y4 goes here
BestFit	Description of BestFit goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPerspective')} **Related Topics**

CorelScript.EffectPinchPunch

Sub **EffectPinchPunch**(ByVal **Level** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPinchPunch')} **Related Topics**

CorelScript.EffectPixelate

Sub **EffectPixelate**(ByVal **Width** As Long, ByVal **Height** As Long, ByVal **Opacity** As Long, ByVal **Mode** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here
Opacity	Description of Opacity goes here
Mode	Description of Mode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPixelate')} **Related Topics**

CorelScript.EffectPlasterWall

Sub **EffectPlasterWall**(ByVal **Detail** As Long, ByVal **Variation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Detail	Description of Detail goes here
Variation	Description of Variation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPlasterWall')} **Related Topics**

CorelScript.EffectPlastic

Sub **EffectPlastic**(ByVal **Highlight** As Long, ByVal **Depth** As Long, ByVal **Smoothness** As Long, ByVal **Direction** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Highlight	Description of Highlight goes here
Depth	Description of Depth goes here
Smoothness	Description of Smoothness goes here
Direction	Description of Direction goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPlastic')} **Related Topics**

CorelScript.EffectPlugin

Sub **EffectPlugin**(ByVal **GroupName** As String, ByVal **EffectName** As String, ByVal **MemoryType** As Long, ByVal **MemorySize** As Long, ByVal **Parameters** As String)

Member of **CorelScript**

Parameters	Description
GroupName	Description of GroupName goes here
EffectName	Description of EffectName goes here
MemoryType	Description of MemoryType goes here
MemorySize	Description of MemorySize goes here
Parameters	Description of Parameters goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPlugin')} **Related Topics**

CorelScript.EffectPointillist

Sub **EffectPointillist**(ByVal **Size** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPointillist')} **Related Topics**

CorelScript.EffectPsychedelic

Sub **EffectPsychedelic**(ByVal **Level** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPsychedelic')} **Related Topics**

CorelScript.EffectPuzzle

Sub **EffectPuzzle**(ByVal **Width** As Long, ByVal **Height** As Long, ByVal **Offset** As Long, ByVal **Fill** As Long, ByVal **RandSeed** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here
Offset	Description of Offset goes here
Fill	Description of Fill goes here
RandSeed	Description of RandSeed goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectPuzzle')} **Related Topics**

CorelScript.EffectRadialBlur

Sub **EffectRadialBlur**(ByVal **Radius** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectRadialBlur')} **Related Topics**

CorelScript.EffectReliefSculpture

Sub **EffectReliefSculpture**(ByVal **Detail** As Long, ByVal **Depth** As Long, ByVal **Smoothness** As Long, ByVal **Direction** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Detail	Description of Detail goes here
Depth	Description of Depth goes here
Smoothness	Description of Smoothness goes here
Direction	Description of Direction goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectReliefSculpture')} **Related Topics**

CorelScript.EffectRemoveMoire

Sub **EffectRemoveMoire**(ByVal **Level** As Long, ByVal **Mode** As Long, ByVal **FinalRes** As Long, ByVal **OriginalRes** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here
Mode	Description of Mode goes here
FinalRes	Description of FinalRes goes here
OriginalRes	Description of OriginalRes goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectRemoveMoire')} **Related Topics**

CorelScript.EffectRemoveNoise

Sub **EffectRemoveNoise**(ByVal **Threshold** As Long, ByVal **Auto** As Boolean)

Member of **CorelScript**

Parameters	Description
Threshold	Description of Threshold goes here
Auto	Description of Auto goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectRemoveNoise')} **Related Topics**

CorelScript.EffectRipple

Sub **EffectRipple**(ByVal **Period** As Long, ByVal **Amplitude** As Long, ByVal **Angle** As Long, ByVal **Distort** As Boolean, ByVal **Mode** As Long)

Member of **CorelScript**

Parameters	Description
Period	Description of Period goes here
Amplitude	Description of Amplitude goes here
Angle	Description of Angle goes here
Distort	Description of Distort goes here
Mode	Description of Mode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectRipple')} **Related Topics**

CorelScript.EffectScatter

Sub **EffectScatter**(ByVal **Horizontal** As Long, ByVal **Vertical** As Long)

Member of **CorelScript**

Parameters	Description
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectScatter')} **Related Topics**

CorelScript.EffectScraperboard

Sub **EffectScraperboard**(ByVal **Style** As Long, ByVal **Density** As Long, ByVal **Size** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Density	Description of Density goes here
Size	Description of Size goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectScraperboard')} **Related Topics**

CorelScript.EffectScreenDoor

Sub **EffectScreenDoor**(ByVal **Background** As Long, ByVal **MeshDensity** As Long, ByVal **Softness** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Background	Description of Background goes here
MeshDensity	Description of MeshDensity goes here
Softness	Description of Softness goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectScreenDoor')} **Related Topics**

CorelScript.EffectSharpen

Sub **EffectSharpen**(ByVal **EdgeLevel** As Long, ByVal **Threshold** As Long, ByVal **Intensity** As Boolean)

Member of **CorelScript**

Parameters	Description
EdgeLevel	Description of EdgeLevel goes here
Threshold	Description of Threshold goes here
Intensity	Description of Intensity goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSharpen')} **Related Topics**

CorelScript.EffectShear

Sub **EffectShear**(ByVal **Scale** As Long, ByVal **Border** As Long, ByVal **Orientation** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Scale	Description of Scale goes here
Border	Description of Border goes here
Orientation	Description of Orientation goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectShear')} **Related Topics**

CorelScript.EffectShearTable

Sub **EffectShearTable**(ByVal **Number** As Long, ByVal **Value** As Long)

Member of **CorelScript**

Parameters	Description
Number	Description of Number goes here
Value	Description of Value goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectShearTable')} **Related Topics**

CorelScript.EffectSketchPad

Sub **EffectSketchPad**(ByVal **Type** As Long, ByVal **Style** As Long, ByVal **Outline** As Long, ByVal **Lead** As Long)

Member of **CorelScript**

Parameters	Description
Type	Description of Type goes here
Style	Description of Style goes here
Outline	Description of Outline goes here
Lead	Description of Lead goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSketchPad')} **Related Topics**

CorelScript.EffectSmokedGlass

Sub **EffectSmokedGlass**(ByVal **Tint** As Long, ByVal **Percent** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Tint	Description of Tint goes here
Percent	Description of Percent goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSmokedGlass')} **Related Topics**

CorelScript.EffectSmooth

Sub **EffectSmooth**(ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSmooth')} **Related Topics**

CorelScript.EffectSoften

Sub **EffectSoften**(ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSoften')} **Related Topics**

CorelScript.EffectSolarize

Sub **EffectSolarize**(ByVal **Level** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSolarize')} **Related Topics**

CorelScript.EffectSphere

Sub **EffectSphere**(ByVal **Percentage** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Member of **CorelScript**

Parameters	Description
Percentage	Description of Percentage goes here
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSphere')} **Related Topics**

CorelScript.EffectStainedGlass

Sub **EffectStainedGlass**(ByVal **Size** As Long, ByVal **Thickness** As Long, ByVal **Lighting** As Boolean, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Thickness	Description of Thickness goes here
Lighting	Description of Lighting goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectStainedGlass')} **Related Topics**

CorelScript.EffectStone

Sub **EffectStone**(ByVal **Roughness** As Long, ByVal **Detail** As Long, ByVal **Invert** As Boolean, ByVal **Direction** As Long)

Member of **CorelScript**

Parameters	Description
Roughness	Description of Roughness goes here
Detail	Description of Detail goes here
Invert	Description of Invert goes here
Direction	Description of Direction goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectStone')} **Related Topics**

CorelScript.EffectSwirl

Sub **EffectSwirl**(ByVal **Angle** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Member of **CorelScript**

Parameters	Description
Angle	Description of Angle goes here
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectSwirl')} **Related Topics**

CorelScript.EffectTheBoss

Sub **EffectTheBoss**(ByVal **Angle** As Long, ByVal **BevelWidth** As Long, ByVal **BevelHeight** As Long, ByVal **Brightness** As Long, ByVal **Direction** As Long, ByVal **Dropoff** As Long, ByVal **Sharpness** As Long, ByVal **Smooth** As Long, ByVal **Invert** As Boolean)

Member of **CorelScript**

Parameters	Description
Angle	Description of Angle goes here
BevelWidth	Description of BevelWidth goes here
BevelHeight	Description of BevelHeight goes here
Brightness	Description of Brightness goes here
Direction	Description of Direction goes here
Dropoff	Description of Dropoff goes here
Sharpness	Description of Sharpness goes here
Smooth	Description of Smooth goes here
Invert	Description of Invert goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectTheBoss')} **Related Topics**

CorelScript.EffectTile

Sub **EffectTile**(ByVal **Horizontal** As Long, ByVal **Vertical** As Long)

Member of **CorelScript**

Parameters	Description
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectTile')} **Related Topics**

CorelScript.EffectTool

Sub **EffectTool**(ByVal **BrushID** As Long, ByVal **TypeID** As Long, ByVal **MergeMode** As Long, ByVal **Amount** As Long, ByVal **NibShape** As Long, ByVal **Size** As Long, ByVal **Transparency** As Long, ByVal **Rotate** As Long, ByVal **Flatten** As Long, ByVal **SoftEdge** As Long)

Member of **CorelScript**

Parameters	Description
BrushID	Description of BrushID goes here
TypeID	Description of TypeID goes here
MergeMode	Description of MergeMode goes here
Amount	Description of Amount goes here
NibShape	Description of NibShape goes here
Size	Description of Size goes here
Transparency	Description of Transparency goes here
Rotate	Description of Rotate goes here
Flatten	Description of Flatten goes here
SoftEdge	Description of SoftEdge goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectTool')} **Related Topics**

CorelScript.EffectTraceContour

Sub **EffectTraceContour**(ByVal **Level** As Long, ByVal **EdgeType** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here
EdgeType	Description of EdgeType goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectTraceContour')} **Related Topics**

CorelScript.EffectUnderpainting

Sub **EffectUnderpainting**(ByVal **Amount** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Amount	Description of Amount goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectUnderpainting')} **Related Topics**

CorelScript.EffectUnsharpMask

Sub **EffectUnsharpMask**(ByVal **Radius** As Long, ByVal **Percentage** As Long, ByVal **Threshold** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
Percentage	Description of Percentage goes here
Threshold	Description of Threshold goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectUnsharpMask')} **Related Topics**

CorelScript.EffectUserDefined

Sub **EffectUserDefined**(ByVal **Divisor** As Long, ByVal **Offset** As Long, ByVal **Intensity** As Boolean)

Member of **CorelScript**

Parameters	Description
Divisor	Description of Divisor goes here
Offset	Description of Offset goes here
Intensity	Description of Intensity goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectUserDefined')} **Related Topics**

CorelScript.EffectUserDefinedPoint

Sub **EffectUserDefinedPoint**(ByVal **Index** As Long, ByVal **Value** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
Value	Description of Value goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectUserDefinedPoint')} **Related Topics**

CorelScript.EffectVignette

Sub **EffectVignette**(ByVal **Shape** As Long, ByVal **Offset** As Long, ByVal **Fade** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Shape	Description of Shape goes here
Offset	Description of Offset goes here
Fade	Description of Fade goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectVignette')} **Related Topics**

CorelScript.EffectVortex

Sub **EffectVortex**(ByVal **Style** As Long, ByVal **Inner** As Long, ByVal **Outer** As Long, ByVal **Size** As Long, ByVal **XCenter** As Long, ByVal **YCenter** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
Inner	Description of Inner goes here
Outer	Description of Outer goes here
Size	Description of Size goes here
XCenter	Description of XCenter goes here
YCenter	Description of YCenter goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectVortex')} **Related Topics**

CorelScript.EffectWatercolor

Sub **EffectWatercolor**(ByVal **Size** As Long, ByVal **Bleed** As Long, ByVal **Granulation** As Long, ByVal **WaterAmount** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Bleed	Description of Bleed goes here
Granulation	Description of Granulation goes here
WaterAmount	Description of WaterAmount goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWatercolor')} **Related Topics**

CorelScript.EffectWaterMarker

Sub **EffectWaterMarker**(ByVal **Mode** As Long, ByVal **Coloration** As Long, ByVal **Size** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here
Coloration	Description of Coloration goes here
Size	Description of Size goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWaterMarker')} **Related Topics**

CorelScript.EffectWavePaper

Sub **EffectWavePaper**(ByVal **Pressure** As Long, ByVal **ColorMode** As Long)

Member of **CorelScript**

Parameters	Description
Pressure	Description of Pressure goes here
ColorMode	Description of ColorMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWavePaper')} **Related Topics**

CorelScript.EffectWeather

Sub **EffectWeather**(ByVal **Forecast** As Long, ByVal **Strength** As Long, ByVal **Size** As Long, ByVal **Variation** As Long, ByVal **Direction** As Long)

Member of **CorelScript**

Parameters	Description
Forecast	Description of Forecast goes here
Strength	Description of Strength goes here
Size	Description of Size goes here
Variation	Description of Variation goes here
Direction	Description of Direction goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWeather')} **Related Topics**

CorelScript.EffectWetPaint

Sub **EffectWetPaint**(ByVal **Wetness** As Long, ByVal **Percentage** As Long)

Member of **CorelScript**

Parameters	Description
Wetness	Description of Wetness goes here
Percentage	Description of Percentage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWetPaint')} **Related Topics**

CorelScript.EffectWhirlpool

Sub **EffectWhirlpool**(ByVal **Spacing** As Long, ByVal **Smear** As Long, ByVal **Twist** As Long, ByVal **Streak** As Long, ByVal **Warp** As Boolean)

Member of **CorelScript**

Parameters	Description
Spacing	Description of Spacing goes here
Smear	Description of Smear goes here
Twist	Description of Twist goes here
Streak	Description of Streak goes here
Warp	Description of Warp goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWhirlpool')} **Related Topics**

CorelScript.EffectWind

Sub **EffectWind**(ByVal **Strength** As Long, ByVal **Opacity** As Long, ByVal **Direction** As Long)

Member of **CorelScript**

Parameters	Description
Strength	Description of Strength goes here
Opacity	Description of Opacity goes here
Direction	Description of Direction goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectWind')} **Related Topics**

CorelScript.EffectZigZag

Sub **EffectZigZag**(ByVal **Waves** As Long, ByVal **Strength** As Long, ByVal **Damping** As Long, ByVal **Type** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Member of **CorelScript**

Parameters	Description
Waves	Description of Waves goes here
Strength	Description of Strength goes here
Damping	Description of Damping goes here
Type	Description of Type goes here
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectZigZag')} **Related Topics**

CorelScript.EffectZoom

Sub **EffectZoom**(ByVal **Radius** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EffectZoom')} **Related Topics**

CorelScript.Ellipse

Sub **Ellipse**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Ellipse')} **Related Topics**

CorelScript.EllipseTool

Sub **EllipseTool**(ByVal **Width** As Long, ByVal **Transparency** As Long, ByVal **MergeMode** As Long, ByVal **AntiAlias** As Boolean, ByVal **RenderObject** As Boolean, ByVal **Fill** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Transparency	Description of Transparency goes here
MergeMode	Description of MergeMode goes here
AntiAlias	Description of AntiAlias goes here
RenderObject	Description of RenderObject goes here
Fill	Description of Fill goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EllipseTool')} **Related Topics**

CorelScript.EndAdjustEffect

Sub **EndAdjustEffect**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndAdjustEffect')} **Related Topics**

CorelScript.EndColorEffect

Sub **EndColorEffect**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndColorEffect')} **Related Topics**

CorelScript.EndColorMask

Sub **EndColorMask**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndColorMask')} **Related Topics**

CorelScript.EndColorReplace

Sub **EndColorReplace**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndColorReplace')} **Related Topics**

CorelScript.EndColorTable

Sub **EndColorTable**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndColorTable')} **Related Topics**

CorelScript.EndConvertDuotone

Sub **EndConvertDuotone**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndConvertDuotone')} **Related Topics**

CorelScript.EndConvertPaletted

Sub **EndConvertPaletted**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndConvertPaletted')} **Related Topics**

CorelScript.EndDraw

Sub **EndDraw**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndDraw')} **Related Topics**

CorelScript.EndEditFill

Sub **EndEditFill**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndEditFill')} **Related Topics**

CorelScript.EndEffectFrame

Sub **EndEffectFrame**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndEffectFrame')} **Related Topics**

CorelScript.EndEffectLighting

Sub **EndEffectLighting**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndEffectLighting')} **Related Topics**

CorelScript.EndEffectMeshWarp

Sub **EndEffectMeshWarp**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndEffectMeshWarp')} **Related Topics**

CorelScript.EndEffectShear

Sub **EndEffectShear**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndEffectShear')} **Related Topics**

CorelScript.EndEffectUserDefined

Sub **EndEffectUserDefined**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndEffectUserDefined')} **Related Topics**

CorelScript.EndGuideline

Sub **EndGuideline**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndGuideline')} **Related Topics**

CorelScript.EndImageEqualize

Sub **EndImageEqualize**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndImageEqualize')} **Related Topics**

CorelScript.EndImageSTBalance

Sub **EndImageSTBalance**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndImageSTBalance')} **Related Topics**

CorelScript.EndImageToneCurve

Sub **EndImageToneCurve**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndImageToneCurve')} **Related Topics**

CorelScript.EndMaskCreate

Sub **EndMaskCreate**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndMaskCreate')} **Related Topics**

CorelScript.EndMovieFrameRate

Sub **EndMovieFrameRate**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndMovieFrameRate')} **Related Topics**

CorelScript.EndObject

Sub **EndObject**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndObject')} **Related Topics**

CorelScript.EndObjectColorTransparencyTool

Sub **EndObjectColorTransparencyTool**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndObjectColorTransparencyTool')} **Related Topics**

CorelScript.EndObjectTagWWWURL

Sub **EndObjectTagWWWURL**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_EndObjectTagWWWURL')} **Related Topics**

CorelScript.Eraser

Sub **Eraser**(ByVal **Width** As Long, ByVal **Flatten** As Long, ByVal **Rotate** As Long, ByVal **NibShape** As Long, ByVal **Transparency** As Long, ByVal **SoftEdge** As Long, ByVal **AntiAlias** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Flatten	Description of Flatten goes here
Rotate	Description of Rotate goes here
NibShape	Description of NibShape goes here
Transparency	Description of Transparency goes here
SoftEdge	Description of SoftEdge goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Eraser')} **Related Topics**

CorelScript.FadeLastCommand

Sub **FadeLastCommand**(ByVal **Percent** As Long, ByVal **MergeMode** As Long)

Member of **CorelScript**

Parameters	Description
Percent	Description of Percent goes here
MergeMode	Description of MergeMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FadeLastCommand')} **Related Topics**

CorelScript.FileClose

Sub **FileClose**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileClose')} **Related Topics**

CorelScript.FileExit

Sub **FileExit**(ByVal **PromptUser** As Boolean)

Member of **CorelScript**

Parameters	Description
PromptUser	Description of PromptUser goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileExit')} **Related Topics**

CorelScript.FileImport

Sub **FileImport**(ByVal **FileName** As String, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **LoadType** As Long, ByVal **X** As Long, ByVal **Y** As Long, ByVal **BoxLeft** As Long, ByVal **BoxTop** As Long, ByVal **BoxRight** As Long, ByVal **BoxBottom** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
LoadType	Description of LoadType goes here
X	Description of X goes here
Y	Description of Y goes here
BoxLeft	Description of BoxLeft goes here
BoxTop	Description of BoxTop goes here
BoxRight	Description of BoxRight goes here
BoxBottom	Description of BoxBottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileImport')} **Related Topics**

CorelScript.FileNew

Sub **FileNew**(ByVal **Width** As Long, ByVal **Height** As Long, ByVal **Type** As Long, ByVal **HRes** As Long, ByVal **VRes** As Long, ByVal **PartialFile** As Boolean, ByVal **MovieFile** As Boolean, ByVal **NumberFrames** As Long, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **Background** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here
Type	Description of Type goes here
HRes	Description of HRes goes here
VRes	Description of VRes goes here
PartialFile	Description of PartialFile goes here
MovieFile	Description of MovieFile goes here
NumberFrames	Description of NumberFrames goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
Background	Description of Background goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileNew')} **Related Topics**

CorelScript.FileOpen

Sub **FileOpen**(ByVal **FileName** As String, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **LoadType** As Long, ByVal **StartFrame** As Long, ByVal **EndFrame** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
LoadType	Description of LoadType goes here
StartFrame	Description of StartFrame goes here
EndFrame	Description of EndFrame goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileOpen')} **Related Topics**

CorelScript.FilePrint

Sub **FilePrint**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilePrint')} **Related Topics**

CorelScript.FileRevert

Sub **FileRevert**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileRevert')} **Related Topics**

CorelScript.FileSave

Sub **FileSave**(ByVal **FileName** As String, ByVal **FilterID** As Long, ByVal **Compression** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
FilterID	Description of FilterID goes here
Compression	Description of Compression goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileSave')} **Related Topics**

CorelScript.FileSelectPartialArea

Sub **FileSelectPartialArea**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FileSelectPartialArea')} **Related Topics**

CorelScript.Fill

Sub **Fill**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Fill')} **Related Topics**

CorelScript.FillBitmap

Sub **FillBitmap**(ByVal **BitmapName** As String, ByVal **Width** As Long, ByVal **Height** As Long, ByVal **Xoffset** As Long, ByVal **Yoffset** As Long, ByVal **TileColumn** As Boolean, ByVal **TileOffset** As Long, ByVal **MaintainAspect** As Boolean, ByVal **Scale** As Boolean, ByVal **OriginalSize** As Boolean, ByVal **SkewAngle** As Long, ByVal **RotateAngle** As Long)

Member of **CorelScript**

Parameters	Description
BitmapName	Description of BitmapName goes here
Width	Description of Width goes here
Height	Description of Height goes here
Xoffset	Description of Xoffset goes here
Yoffset	Description of Yoffset goes here
TileColumn	Description of TileColumn goes here
TileOffset	Description of TileOffset goes here
MaintainAspect	Description of MaintainAspect goes here
Scale	Description of Scale goes here
OriginalSize	Description of OriginalSize goes here
SkewAngle	Description of SkewAngle goes here
RotateAngle	Description of RotateAngle goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillBitmap')} **Related Topics**

CorelScript.FillFountain

Sub **FillFountain**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillFountain')} **Related Topics**

CorelScript.FillFountainApply

Sub **FillFountainApply**(ByVal **Type** As Long, ByVal **Colors** As Long, ByVal **Steps** As Long, ByVal **Angle** As Long, ByVal **EdgePad** As Long, ByVal **HorizontalOffset** As Long, ByVal **VerticalOffset** As Long, ByVal **Midpoint** As Long)

Member of **CorelScript**

Parameters	Description
Type	Description of Type goes here
Colors	Description of Colors goes here
Steps	Description of Steps goes here
Angle	Description of Angle goes here
EdgePad	Description of EdgePad goes here
HorizontalOffset	Description of HorizontalOffset goes here
VerticalOffset	Description of VerticalOffset goes here
Midpoint	Description of Midpoint goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillFountainApply')} **Related Topics**

CorelScript.FillFountainColor

Sub **FillFountainColor**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **Position** As Long, ByVal **Index** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
Position	Description of Position goes here
Index	Description of Index goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillFountainColor')} **Related Topics**

CorelScript.FillSolid

Sub **FillSolid**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillSolid')} **Related Topics**

CorelScript.FillTexture

Sub **FillTexture**(ByVal **LibraryName** As String, ByVal **TextureName** As String, ByVal **StyleName** As String)

Member of **CorelScript**

Parameters	Description
LibraryName	Description of LibraryName goes here
TextureName	Description of TextureName goes here
StyleName	Description of StyleName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillTexture')} **Related Topics**

CorelScript.FillTextureSettings

Sub **FillTextureSettings**(ByVal **Index** As Long, ByVal **Value1** As Long, ByVal **Value2** As Long, ByVal **Value3** As Long, ByVal **Value4** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
Value1	Description of Value1 goes here
Value2	Description of Value2 goes here
Value3	Description of Value3 goes here
Value4	Description of Value4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillTextureSettings')} **Related Topics**

CorelScript.FillTool

Sub **FillTool**(ByVal **Transparency** As Long, ByVal **MergeMode** As Long)

Member of **CorelScript**

Parameters	Description
Transparency	Description of Transparency goes here
MergeMode	Description of MergeMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FillTool')} **Related Topics**

CorelScript.FilterAVI

Sub **FilterAVI**(ByVal **Handler** As Long, ByVal **KeyFrame** As Long, ByVal **Quality** As Long, ByVal **BPS** As Long, ByVal **Flags** As Long, ByVal **Interleave** As Long)

Member of **CorelScript**

Parameters	Description
Handler	Description of Handler goes here
KeyFrame	Description of KeyFrame goes here
Quality	Description of Quality goes here
BPS	Description of BPS goes here
Flags	Description of Flags goes here
Interleave	Description of Interleave goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterAVI')} **Related Topics**

CorelScript.FilterFPX

Sub **FilterFPX**(ByVal **Compression** As Long, ByVal **Decimation** As Long, ByVal **Quality** As Long)

Member of **CorelScript**

Parameters	Description
Compression	Description of Compression goes here
Decimation	Description of Decimation goes here
Quality	Description of Quality goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterFPX')} **Related Topics**

CorelScript.FilterGIF

Sub **FilterGIF**(ByVal **InvertMask** As Long, ByVal **Interlace** As Long, ByVal **Transparent** As Long, ByVal **Index** As Long, ByVal **Delay** As Long, ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long)

Member of **CorelScript**

Parameters	Description
InvertMask	Description of InvertMask goes here
Interlace	Description of Interlace goes here
Transparent	Description of Transparent goes here
Index	Description of Index goes here
Delay	Description of Delay goes here
Red	Description of Red goes here
Green	Description of Green goes here
Blue	Description of Blue goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterGIF')} **Related Topics**

CorelScript.FilterICO

Sub **FilterICO**(ByVal **Transparent** As Long, ByVal **Inverse** As Long)

Member of **CorelScript**

Parameters	Description
Transparent	Description of Transparent goes here
Inverse	Description of Inverse goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterICO')} **Related Topics**

CorelScript.FilterJPG

Sub **FilterJPG**(ByVal **Quality** As Long, ByVal **Progressive** As Long, ByVal **Smoothing** As Long, ByVal **SubFormat** As Long, ByVal **Optimized** As Long)

Member of **CorelScript**

Parameters	Description
Quality	Description of Quality goes here
Progressive	Description of Progressive goes here
Smoothing	Description of Smoothing goes here
SubFormat	Description of SubFormat goes here
Optimized	Description of Optimized goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterJPG')} **Related Topics**

CorelScript.FilterMPG

Sub **FilterMPG**(ByVal **Emphasis** As Long, ByVal **Quality** As Long, ByVal **Audio** As Boolean, ByVal **Video** As Boolean, ByVal **System** As Boolean)

Member of **CorelScript**

Parameters	Description
Emphasis	Description of Emphasis goes here
Quality	Description of Quality goes here
Audio	Description of Audio goes here
Video	Description of Video goes here
System	Description of System goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterMPG')} **Related Topics**

CorelScript.FilterOS2

Sub **FilterOS2**(ByVal **Format** As Long)

Member of **CorelScript**

Parameters	Description
Format	Description of Format goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterOS2')} **Related Topics**

CorelScript.FilterPCD

Sub **FilterPCD**(ByVal **Resolution** As Long, ByVal **Color** As Long)

Member of **CorelScript**

Parameters	Description
Resolution	Description of Resolution goes here
Color	Description of Color goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterPCD')} **Related Topics**

CorelScript.FilterPNG

Sub **FilterPNG**(ByVal **Interlace** As Long)

Member of **CorelScript**

Parameters	Description
Interlace	Description of Interlace goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterPNG')} **Related Topics**

CorelScript.FilterRAW

Sub **FilterRAW**(ByVal **Format** As Long, ByVal **Type** As Long, ByVal **Width** As Long, ByVal **Height** As Long, ByVal **BitCount** As Long, ByVal **Header** As Long, ByVal **Mask** As Long, ByVal **PaletteType** As Long, ByVal **Location** As Long, ByVal **Entries** As Long)

Member of **CorelScript**

Parameters	Description
Format	Description of Format goes here
Type	Description of Type goes here
Width	Description of Width goes here
Height	Description of Height goes here
BitCount	Description of BitCount goes here
Header	Description of Header goes here
Mask	Description of Mask goes here
PaletteType	Description of PaletteType goes here
Location	Description of Location goes here
Entries	Description of Entries goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterRAW')} **Related Topics**

CorelScript.FilterTGA

Sub **FilterTGA**(ByVal **Format** As Long)

Member of **CorelScript**

Parameters	Description
Format	Description of Format goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterTGA')} **Related Topics**

CorelScript.FilterWVL

Sub **FilterWVL**(ByVal **Quality** As Long, ByVal **Speed** As Long, ByVal **Path** As Long, ByVal **Contrast** As Long, ByVal **Edge** As Long)

Member of **CorelScript**

Parameters	Description
Quality	Description of Quality goes here
Speed	Description of Speed goes here
Path	Description of Path goes here
Contrast	Description of Contrast goes here
Edge	Description of Edge goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_FilterWVL')} **Related Topics**

CorelScript.GetChannelCount

Function **GetChannelCount**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetChannelCount')} **Related Topics**

CorelScript.GetChannelName

Function **GetChannelName**(ByVal **ChannelID** As Long) As String

Member of **CorelScript**

Parameters	Description
ChannelID	Description of ChannelID goes here

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetChannelName')} **Related Topics**

CorelScript.GetCurrentMovieFrame

Function **GetCurrentMovieFrame**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetCurrentMovieFrame')} **Related Topics**

CorelScript.GetDocumentCount

Function **GetDocumentCount**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentCount')} **Related Topics**

CorelScript.GetDocumentHeight

Function **GetDocumentHeight**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentHeight')} **Related Topics**

CorelScript.GetDocumentHRes

Function **GetDocumentHRes**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentHRes')} **Related Topics**

CorelScript.GetDocumentIsMovie

Function **GetDocumentIsMovie**() As Boolean

Member of **CorelScript**

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentIsMovie')} **Related Topics**

CorelScript.GetDocumentIsPartial

Function **GetDocumentIsPartial**() As Boolean

Member of **CorelScript**

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentIsPartial')} **Related Topics**

CorelScript.GetDocumentModified

Function **GetDocumentModified**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentModified')} **Related Topics**

CorelScript.GetDocumentName

Function **GetDocumentName**() As String

Member of **CorelScript**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentName')} **Related Topics**

CorelScript.GetDocumentType

Function **GetDocumentType**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentType')} **Related Topics**

CorelScript.GetDocumentVRes

Function **GetDocumentVRes**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentVRes')} **Related Topics**

CorelScript.GetDocumentWidth

Function **GetDocumentWidth**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentWidth')} **Related Topics**

CorelScript.GetDocumentXdpi

Function **GetDocumentXdpi**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentXdpi')} **Related Topics**

CorelScript.GetDocumentXGridFrequency

Function **GetDocumentXGridFrequency**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentXGridFrequency')} **Related Topics**

CorelScript.GetDocumentXRulerUnits

Function **GetDocumentXRulerUnits**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentXRulerUnits')} **Related Topics**

CorelScript.GetDocumentYdpi

Function **GetDocumentYdpi**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentYdpi')} **Related Topics**

CorelScript.GetDocumentYGridFrequency

Function **GetDocumentYGridFrequency**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentYGridFrequency')} **Related Topics**

CorelScript.GetDocumentYRulerUnits

Function **GetDocumentYRulerUnits**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetDocumentYRulerUnits')} **Related Topics**

CorelScript.GetFillColor

Function **GetFillColor**(ByRef **ColorModel** As Long, ByRef **Color1** As Long, ByRef **Color2** As Long, ByRef **Color3** As Long, ByRef **Color4** As Long) As Boolean

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetFillColor')} **Related Topics**

CorelScript.GetFrameCount

Function **GetFrameCount**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetFrameCount')} **Related Topics**

CorelScript.GetGuidelineCount

Function **GetGuidelineCount**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetGuidelineCount')} **Related Topics**

CorelScript.GetGuidelineProperties

Sub **GetGuidelineProperties**(ByVal **Index** As Long, ByRef **X** As Long, ByRef **Y** As Long, ByRef **Angle** As Long, ByRef **Horizontal** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
X	Description of X goes here
Y	Description of Y goes here
Angle	Description of Angle goes here
Horizontal	Description of Horizontal goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetGuidelineProperties')} **Related Topics**

CorelScript.GetMaskPresent

Function **GetMaskPresent**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetMaskPresent')} **Related Topics**

CorelScript.GetMaskRectangle

Sub **GetMaskRectangle**(ByRef **Left** As Long, ByRef **Top** As Long, ByRef **Right** As Long, ByRef **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetMaskRectangle')} **Related Topics**

CorelScript.GetObjectCount

Function **GetObjectCount**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectCount')} **Related Topics**

CorelScript.GetObjectIsEditable

Function **GetObjectIsEditable**(ByVal **ObjectID** As Long) As Boolean

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectIsEditable')} **Related Topics**

CorelScript.GetObjectIsSelected

Function **GetObjectIsSelected**(ByVal **ObjectID** As Long) As Boolean

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectIsSelected')} **Related Topics**

CorelScript.GetObjectIsVisible

Function **GetObjectIsVisible**(ByVal **ObjectID** As Long) As Boolean

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectIsVisible')} **Related Topics**

CorelScript.GetObjectMergeMode

Function **GetObjectMergeMode**(ByVal **ObjectID** As Long) As Long

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectMergeMode')} **Related Topics**

CorelScript.GetObjectName

Function **GetObjectName**(ByVal **ObjectID** As Long) As String

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectName')} **Related Topics**

CorelScript.GetObjectOpacity

Function **GetObjectOpacity**(ByVal **ObjectID** As Long) As Long

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectOpacity')} **Related Topics**

CorelScript.GetObjectProperties

Sub **GetObjectProperties**(ByVal **ObjectID** As Long, ByVal **Name** As String, ByRef **Opacity** As Long, ByRef **MergeMode** As Long, ByRef **Visible** As Boolean, ByRef **Clipped** As Boolean, ByRef **Enabled** As Boolean, ByRef **Linked** As Boolean)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Name	Description of Name goes here
Opacity	Description of Opacity goes here
MergeMode	Description of MergeMode goes here
Visible	Description of Visible goes here
Clipped	Description of Clipped goes here
Enabled	Description of Enabled goes here
Linked	Description of Linked goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectProperties')} **Related Topics**

CorelScript.GetObjectRectangle

Sub **GetObjectRectangle**(ByVal **ObjectID** As Long, ByRef **Left** As Long, ByRef **Top** As Long, ByRef **Right** As Long, ByRef **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetObjectRectangle')} **Related Topics**

CorelScript.GetPaintColor

Sub **GetPaintColor**(ByRef **ColorModel** As Long, ByRef **Color1** As Long, ByRef **Color2** As Long, ByRef **Color3** As Long, ByRef **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPaintColor')} **Related Topics**

CorelScript.GetPaintVersion

Function **GetPaintVersion**() As String

Member of **CorelScript**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPaintVersion')} **Related Topics**

CorelScript.GetPaperColor

Sub **GetPaperColor**(ByRef **ColorModel** As Long, ByRef **Color1** As Long, ByRef **Color2** As Long, ByRef **Color3** As Long, ByRef **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPaperColor')} **Related Topics**

CorelScript.GetPartialDocumentHeight

Function **GetPartialDocumentHeight**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPartialDocumentHeight')} **Related Topics**

CorelScript.GetPartialDocumentWidth

Function **GetPartialDocumentWidth**() As Long

Member of **CorelScript**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPartialDocumentWidth')} **Related Topics**

CorelScript.GetPhotoPaintDir

Function **GetPhotoPaintDir**() As String

Member of **CorelScript**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPhotoPaintDir')} **Related Topics**

CorelScript.GetPixelColor

Function **GetPixelColor**(ByVal **X** As Long, ByVal **Y** As Long, ByRef **ColorModel** As Long, ByRef **Color1** As Long, ByRef **Color2** As Long, ByRef **Color3** As Long, ByRef **Color4** As Long) As Boolean

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetPixelColor')} **Related Topics**

CorelScript.GetSelectedObjectsRectangle

Sub **GetSelectedObjectsRectangle**(ByRef **Left** As Long, ByRef **Top** As Long, ByRef **Right** As Long, ByRef **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GetSelectedObjectsRectangle')} **Related Topics**

CorelScript.Gradient

Sub **Gradient**(ByVal **Type** As Long, ByVal **MarkerPos** As Long)

Member of **CorelScript**

Parameters	Description
Type	Description of Type goes here
MarkerPos	Description of MarkerPos goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Gradient')} **Related Topics**

CorelScript.GradientPoint

Sub **GradientPoint**(ByVal **Index** As Long, ByVal **PtX** As Long, ByVal **PtY** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **Transparency** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
PtX	Description of PtX goes here
PtY	Description of PtY goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
Transparency	Description of Transparency goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GradientPoint')} **Related Topics**

CorelScript.GradientTool

Sub **GradientTool**(ByVal **Style** As Long, ByVal **MergeMode** As Long, ByVal **Transparency** As Long, ByVal **Handles** As Long)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
MergeMode	Description of MergeMode goes here
Transparency	Description of Transparency goes here
Handles	Description of Handles goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GradientTool')} **Related Topics**

CorelScript.GridSetup

Sub **GridSetup**(ByVal **GridX** As Double, ByVal **GridY** As Double, ByVal **Units** As Long, ByVal **GridOn** As Boolean)

Member of **CorelScript**

Parameters	Description
GridX	Description of GridX goes here
GridY	Description of GridY goes here
Units	Description of Units goes here
GridOn	Description of GridOn goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GridSetup')} **Related Topics**

CorelScript.GuidelineAdd

Sub **GuidelineAdd**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **Angle** As Long, ByVal **Horizontal** As Boolean)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
Angle	Description of Angle goes here
Horizontal	Description of Horizontal goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GuidelineAdd')} **Related Topics**

CorelScript.GuidelineDelete

Sub **GuidelineDelete**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GuidelineDelete')} **Related Topics**

CorelScript.GuidelineMove

Sub **GuidelineMove**(ByVal **X** As Long, ByVal **Y** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GuidelineMove')} **Related Topics**

CorelScript.GuidelineSelect

Sub **GuidelineSelect**(ByVal **GuideID** As Long, ByVal **Selected** As Boolean)

Member of **CorelScript**

Parameters	Description
GuideID	Description of GuideID goes here
Selected	Description of Selected goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_GuidelineSelect')} **Related Topics**

CorelScript.ImageApplyICCProfile

Sub **ImageApplyICCProfile**(ByVal **FileName** As String)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageApplyICCProfile')} **Related Topics**

CorelScript.ImageAutoEqualize

Sub **ImageAutoEqualize**(ByVal **AutoBlack** As Long, ByVal **AutoWhite** As Long)

Member of **CorelScript**

Parameters	Description
AutoBlack	Description of AutoBlack goes here
AutoWhite	Description of AutoWhite goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageAutoEqualize')} **Related Topics**

CorelScript.ImageBCI

Sub **ImageBCI**(ByVal **Brightness** As Long, ByVal **Contrast** As Long, ByVal **Intensity** As Long)

Member of **CorelScript**

Parameters	Description
Brightness	Description of Brightness goes here
Contrast	Description of Contrast goes here
Intensity	Description of Intensity goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageBCI')} **Related Topics**

CorelScript.ImageColorBalance

Sub **ImageColorBalance**(ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long, ByVal **Shadows** As Boolean, ByVal **Midtones** As Boolean, ByVal **Highlights** As Boolean, ByVal **Luminance** As Boolean)

Member of **CorelScript**

Parameters	Description
Red	Description of Red goes here
Green	Description of Green goes here
Blue	Description of Blue goes here
Shadows	Description of Shadows goes here
Midtones	Description of Midtones goes here
Highlights	Description of Highlights goes here
Luminance	Description of Luminance goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageColorBalance')} **Related Topics**

CorelScript.ImageColorCrop

Sub **ImageColorCrop**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **ToleranceMode** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
ToleranceMode	Description of ToleranceMode goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageColorCrop')} **Related Topics**

CorelScript.ImageColorHue

Sub **ImageColorHue**(ByVal **Red** As Long, ByVal **Green** As Long, ByVal **Blue** As Long, ByVal **Shadows** As Boolean, ByVal **Midtones** As Boolean, ByVal **Highlights** As Boolean, ByVal **Luminance** As Boolean)

Member of **CorelScript**

Parameters	Description
Red	Description of Red goes here
Green	Description of Green goes here
Blue	Description of Blue goes here
Shadows	Description of Shadows goes here
Midtones	Description of Midtones goes here
Highlights	Description of Highlights goes here
Luminance	Description of Luminance goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageColorHue')} **Related Topics**

CorelScript.ImageColorTable

Sub **ImageColorTable**(ByVal **Colors** As Long)

Member of **CorelScript**

Parameters	Description
Colors	Description of Colors goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageColorTable')} **Related Topics**

CorelScript.ImageColorTone

Sub **ImageColorTone**(ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Lightness** As Long, ByVal **Brightness** As Long, ByVal **Contrast** As Long, ByVal **Intensity** As Long)

Member of **CorelScript**

Parameters	Description
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Lightness	Description of Lightness goes here
Brightness	Description of Brightness goes here
Contrast	Description of Contrast goes here
Intensity	Description of Intensity goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageColorTone')} **Related Topics**

CorelScript.ImageConvert

Sub **ImageConvert**(ByVal **Type** As Long, ByVal **RenderType** As Long, ByVal **PaletteType** As Long, ByVal **Threshold** As Integer, ByVal **HalftoneType** As Long, ByVal **Angle** As Long, ByVal **Width** As Long, ByVal **Colors** As Long, ByVal **Intensity** As Long, ByVal **Flatten** As Boolean)

Member of **CorelScript**

Parameters	Description
Type	Description of Type goes here
RenderType	Description of RenderType goes here
PaletteType	Description of PaletteType goes here
Threshold	Description of Threshold goes here
HalftoneType	Description of HalftoneType goes here
Angle	Description of Angle goes here
Width	Description of Width goes here
Colors	Description of Colors goes here
Intensity	Description of Intensity goes here
Flatten	Description of Flatten goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageConvert')} **Related Topics**

CorelScript.ImageConvertDuotone

Sub **ImageConvertDuotone**(ByVal **Style** As Long, ByVal **UseOverprints** As Boolean, ByVal **Flatten** As Boolean)

Member of **CorelScript**

Parameters	Description
Style	Description of Style goes here
UseOverprints	Description of UseOverprints goes here
Flatten	Description of Flatten goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageConvertDuotone')} **Related Topics**

CorelScript.ImageConvertPaletted

Sub **ImageConvertPaletted**(ByVal **RenderType** As Long, ByVal **Colors** As Long, ByVal **Intensity** As Long, ByVal **Flatten** As Boolean)

Member of **CorelScript**

Parameters	Description
RenderType	Description of RenderType goes here
Colors	Description of Colors goes here
Intensity	Description of Intensity goes here
Flatten	Description of Flatten goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageConvertPaletted')} **Related Topics**

CorelScript.ImageConvertVideoNTSC

Sub **ImageConvertVideoNTSC**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageConvertVideoNTSC')} **Related Topics**

CorelScript.ImageConvertVideoPAL

Sub **ImageConvertVideoPAL**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageConvertVideoPAL')} **Related Topics**

CorelScript.ImageCrop

Sub **ImageCrop**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageCrop')} **Related Topics**

CorelScript.ImageCropToMask

Sub **ImageCropToMask**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageCropToMask')} **Related Topics**

CorelScript.ImageDeInterlace

Sub **ImageDeInterlace**(ByVal **ReplaceMode** As Long)

Member of **CorelScript**

Parameters	Description
ReplaceMode	Description of ReplaceMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageDeInterlace')} **Related Topics**

CorelScript.ImageDesaturate

Sub **ImageDesaturate**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageDesaturate')} **Related Topics**

CorelScript.ImageDeskew

Sub **ImageDeskew**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageDeskew')} **Related Topics**

CorelScript.ImageDeskewCrop

Sub **ImageDeskewCrop**(ByVal **Angle** As Double, ByVal **Width** As Long, ByVal **Height** As Long, ByVal **PointX** As Long, ByVal **PointY** As Long)

Member of **CorelScript**

Parameters	Description
Angle	Description of Angle goes here
Width	Description of Width goes here
Height	Description of Height goes here
PointX	Description of PointX goes here
PointY	Description of PointY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageDeskewCrop')} **Related Topics**

CorelScript.ImageDuplicate

Sub **ImageDuplicate**(ByVal **FileName** As String, ByVal **MergeObjects** As Boolean)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
MergeObjects	Description of MergeObjects goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageDuplicate')} **Related Topics**

CorelScript.ImageEqualize

Sub **ImageEqualize**(ByVal **Method** As Long, ByVal **AutoBlack** As Long, ByVal **AutoWhite** As Long, ByVal **AutoAdjust** As Boolean)

Member of **CorelScript**

Parameters	Description
Method	Description of Method goes here
AutoBlack	Description of AutoBlack goes here
AutoWhite	Description of AutoWhite goes here
AutoAdjust	Description of AutoAdjust goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageEqualize')} **Related Topics**

CorelScript.ImageEqualizeChannel

Sub **ImageEqualizeChannel**(ByVal **Index** As Long, ByVal **InLow** As Long, ByVal **InHigh** As Long, ByVal **OutLow** As Long, ByVal **OutHigh** As Long, ByVal **Gamma** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
InLow	Description of InLow goes here
InHigh	Description of InHigh goes here
OutLow	Description of OutLow goes here
OutHigh	Description of OutHigh goes here
Gamma	Description of Gamma goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageEqualizeChannel')} **Related Topics**

CorelScript.ImageFlipHorizontal

Sub **ImageFlipHorizontal**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageFlipHorizontal')} **Related Topics**

CorelScript.ImageFlipVertical

Sub **ImageFlipVertical**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageFlipVertical')} **Related Topics**

CorelScript.ImageGamma

Sub **ImageGamma**(ByVal **Value** As Long)

Member of **CorelScript**

Parameters	Description
Value	Description of Value goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageGamma')} **Related Topics**

CorelScript.ImageHSL

Sub **ImageHSL**(ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Lightness** As Long)

Member of **CorelScript**

Parameters	Description
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Lightness	Description of Lightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageHSL')} **Related Topics**

CorelScript.ImageHSLChannel

Sub **ImageHSLChannel**(ByVal **Channel** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Lightness** As Long)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Lightness	Description of Lightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageHSLChannel')} **Related Topics**

CorelScript.ImageInvert

Sub **ImageInvert**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageInvert')} **Related Topics**

CorelScript.ImageLevelThreshold

Sub **ImageLevelThreshold**(ByVal **Channel** As Long, ByVal **Low** As Long, ByVal **Threshold** As Long, ByVal **High** As Long, ByVal **BiLevel** As Long)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here
Low	Description of Low goes here
Threshold	Description of Threshold goes here
High	Description of High goes here
BiLevel	Description of BiLevel goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageLevelThreshold')} **Related Topics**

CorelScript.ImagePapersize

Sub **ImagePapersize**(ByVal **Width** As Long, ByVal **Height** As Long, ByVal **Xoffset** As Long, ByVal **Yoffset** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here
Xoffset	Description of Xoffset goes here
Yoffset	Description of Yoffset goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImagePapersize')} **Related Topics**

CorelScript.ImagePosterize

Sub **ImagePosterize**(ByVal **Level** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImagePosterize')} **Related Topics**

CorelScript.ImageReplaceColors

Sub **ImageReplaceColors**(ByVal **TypeIn** As Long, ByVal **Comp1In** As Long, ByVal **Comp2In** As Long, ByVal **Comp3In** As Long, ByVal **Comp4In** As Long, ByVal **TypeOut** As Long, ByVal **Comp1Out** As Long, ByVal **Comp2Out** As Long, ByVal **Comp3Out** As Long, ByVal **Comp4Out** As Long, ByVal **Range** As Long, ByVal **IgnoreGrayscale** As Boolean, ByVal **SingleColor** As Boolean)

Member of **CorelScript**

Parameters	Description
TypeIn	Description of TypeIn goes here
Comp1In	Description of Comp1In goes here
Comp2In	Description of Comp2In goes here
Comp3In	Description of Comp3In goes here
Comp4In	Description of Comp4In goes here
TypeOut	Description of TypeOut goes here
Comp1Out	Description of Comp1Out goes here
Comp2Out	Description of Comp2Out goes here
Comp3Out	Description of Comp3Out goes here
Comp4Out	Description of Comp4Out goes here
Range	Description of Range goes here
IgnoreGrayscale	Description of IgnoreGrayscale goes here
SingleColor	Description of SingleColor goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageReplaceColors')} **Related Topics**

CorelScript.ImageResample

Sub **ImageResample**(ByVal **Width** As Long, ByVal **Height** As Long, ByVal **HRes** As Long, ByVal **VRes** As Long, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here
HRes	Description of HRes goes here
VRes	Description of VRes goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageResample')} **Related Topics**

CorelScript.ImageRotate

Sub **ImageRotate**(ByVal **Angle** As Double, ByVal **Clip** As Boolean, ByVal **AntiAlias** As Boolean, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Angle	Description of Angle goes here
Clip	Description of Clip goes here
AntiAlias	Description of AntiAlias goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageRotate')} **Related Topics**

CorelScript.ImageSelectiveColor

Sub **ImageSelectiveColor**(ByVal **Mode** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSelectiveColor')} **Related Topics**

CorelScript.ImageSelectiveColorChannel

Sub **ImageSelectiveColorChannel**(ByVal **Channel** As Long, ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long, ByVal **Black** As Long)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here
Cyan	Description of Cyan goes here
Magenta	Description of Magenta goes here
Yellow	Description of Yellow goes here
Black	Description of Black goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSelectiveColorChannel')} **Related Topics**

CorelScript.ImageSetChannel

Sub **ImageSetChannel**(ByVal **Channel** As Long)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSetChannel')} **Related Topics**

CorelScript.ImageSplit

Sub **ImageSplit**(ByVal **Type** As Long)

Member of **CorelScript**

Parameters	Description
Type	Description of Type goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSplit')} **Related Topics**

CorelScript.ImageSprayerList

Sub **ImageSprayerList**(ByVal **List** As String)

Member of **CorelScript**

Parameters	Description
List	Description of List goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSprayerList')} **Related Topics**

CorelScript.ImageSprayerSettings

Sub **ImageSprayerSettings**(ByVal **Dabs** As Long, ByVal **Spacing** As Long, ByVal **Spread** As Long, ByVal **FadeOut** As Long, ByVal **Type** As Long, ByVal **From** As Long, ByVal **To** As Long, ByVal **Start** As Long)

Member of **CorelScript**

Parameters	Description
Dabs	Description of Dabs goes here
Spacing	Description of Spacing goes here
Spread	Description of Spread goes here
FadeOut	Description of FadeOut goes here
Type	Description of Type goes here
From	Description of From goes here
To	Description of To goes here
Start	Description of Start goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSprayerSettings')} **Related Topics**

CorelScript.ImageSprayerTool

Sub **ImageSprayerTool**(ByVal **FileName** As String, ByVal **Row** As Long, ByVal **Column** As Long, ByVal **Size** As Long, ByVal **MergeMode** As Long, ByVal **Transparency** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Row	Description of Row goes here
Column	Description of Column goes here
Size	Description of Size goes here
MergeMode	Description of MergeMode goes here
Transparency	Description of Transparency goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSprayerTool')} **Related Topics**

CorelScript.ImageSTBalance

Sub **ImageSTBalance**(ByVal **Channel** As Long, ByVal **UseLow** As Boolean, ByVal **UseMid** As Boolean, ByVal **UseHigh** As Boolean, ByVal **UseAll** As Boolean)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here
UseLow	Description of UseLow goes here
UseMid	Description of UseMid goes here
UseHigh	Description of UseHigh goes here
UseAll	Description of UseAll goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSTBalance')} **Related Topics**

CorelScript.ImageSTColor

Sub **ImageSTColor**(ByVal **Index** As Long, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Index	Description of Index goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageSTColor')} **Related Topics**

CorelScript.ImageToneChannel

Sub **ImageToneChannel**(ByVal **Channel** As Long, ByVal **Nodes** As Long, ByVal **Gamma** As Long, ByVal **Length** As Long, ByVal **DrawMode** As Long, ByVal **Orientation** As Long)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here
Nodes	Description of Nodes goes here
Gamma	Description of Gamma goes here
Length	Description of Length goes here
DrawMode	Description of DrawMode goes here
Orientation	Description of Orientation goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageToneChannel')} **Related Topics**

CorelScript.ImageToneCurve

Sub **ImageToneCurve**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageToneCurve')} **Related Topics**

CorelScript.ImageTonePoint

Sub **ImageTonePoint**(ByVal **Channel** As Long, ByVal **Index** As Long, ByVal **X** As Long, ByVal **Y** As Long)

Member of **CorelScript**

Parameters	Description
Channel	Description of Channel goes here
Index	Description of Index goes here
X	Description of X goes here
Y	Description of Y goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageTonePoint')} **Related Topics**

CorelScript.ImageToneTable

Sub **ImageToneTable**(ByVal **Number** As Long, ByVal **Curve1** As Long, ByVal **Curve2** As Long, ByVal **Curve3** As Long, ByVal **Curve4** As Long)

Member of **CorelScript**

Parameters	Description
Number	Description of Number goes here
Curve1	Description of Curve1 goes here
Curve2	Description of Curve2 goes here
Curve3	Description of Curve3 goes here
Curve4	Description of Curve4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ImageToneTable')} **Related Topics**

CorelScript.LensCreateFromMask

Sub **LensCreateFromMask**(ByVal **Name** As String)

Member of **CorelScript**

Parameters	Description
Name	Description of Name goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_LensCreateFromMask')} **Related Topics**

CorelScript.LensEdit

Sub **LensEdit**(ByVal **ObjectID** As Long, ByVal **Name** As String)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Name	Description of Name goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_LensEdit')} **Related Topics**

CorelScript.LensNew

Sub **LensNew**(ByVal **Name** As String)

Member of **CorelScript**

Parameters	Description
Name	Description of Name goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_LensNew')} **Related Topics**

CorelScript.LineTool

Sub **LineTool**(ByVal **Width** As Long, ByVal **Transparency** As Long, ByVal **Joints** As Long, ByVal **MergeMode** As Long, ByVal **AntiAlias** As Boolean, ByVal **RenderObject** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Transparency	Description of Transparency goes here
Joints	Description of Joints goes here
MergeMode	Description of MergeMode goes here
AntiAlias	Description of AntiAlias goes here
RenderObject	Description of RenderObject goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_LineTool')} **Related Topics**

CorelScript.LocalUndo

Sub **LocalUndo**(ByVal **Width** As Long, ByVal **Flatten** As Long, ByVal **Rotate** As Long, ByVal **NibShape** As Long, ByVal **Transparency** As Long, ByVal **SoftEdge** As Long, ByVal **AntiAlias** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Flatten	Description of Flatten goes here
Rotate	Description of Rotate goes here
NibShape	Description of NibShape goes here
Transparency	Description of Transparency goes here
SoftEdge	Description of SoftEdge goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_LocalUndo')} **Related Topics**

CorelScript.MaskAffineDistort

Sub **MaskAffineDistort**(ByVal **Xoffset** As Long, ByVal **Yoffset** As Long, ByVal **d11** As Double, ByVal **d12** As Double, ByVal **d21** As Double, ByVal **d22** As Double, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Xoffset	Description of Xoffset goes here
Yoffset	Description of Yoffset goes here
d11	Description of d11 goes here
d12	Description of d12 goes here
d21	Description of d21 goes here
d22	Description of d22 goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskAffineDistort')} **Related Topics**

CorelScript.MaskAlign

Sub **MaskAlign**(ByVal **Horizontal** As Long, ByVal **Vertical** As Long, ByVal **Center** As Boolean, ByVal **Grid** As Boolean, ByVal **ActiveObject** As Boolean, ByVal **SelectedObject** As Boolean, ByVal **Background** As Boolean)

Member of **CorelScript**

Parameters	Description
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here
Center	Description of Center goes here
Grid	Description of Grid goes here
ActiveObject	Description of ActiveObject goes here
SelectedObject	Description of SelectedObject goes here
Background	Description of Background goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskAlign')} **Related Topics**

CorelScript.MaskAntiAlias

Sub **MaskAntiAlias**(ByVal **Tolerance** As Long)

Member of **CorelScript**

Parameters	Description
Tolerance	Description of Tolerance goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskAntiAlias')} **Related Topics**

CorelScript.MaskAverage

Sub **MaskAverage**(ByVal **Radius** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskAverage')} **Related Topics**

CorelScript.MaskBorder

Sub **MaskBorder**(ByVal **Width** As Long, ByVal **Edges** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Edges	Description of Edges goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskBorder')} **Related Topics**

CorelScript.MaskBrush

Sub **MaskBrush**(ByVal **DrawMode** As Long, ByVal **Width** As Long, ByVal **Flatten** As Long, ByVal **Rotate** As Long, ByVal **NibShape** As Long, ByVal **Transparency** As Long, ByVal **SoftEdge** As Long, ByVal **AntiAlias** As Long)

Member of **CorelScript**

Parameters	Description
DrawMode	Description of DrawMode goes here
Width	Description of Width goes here
Flatten	Description of Flatten goes here
Rotate	Description of Rotate goes here
NibShape	Description of NibShape goes here
Transparency	Description of Transparency goes here
SoftEdge	Description of SoftEdge goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskBrush')} **Related Topics**

CorelScript.MaskChannelAdd

Sub **MaskChannelAdd**(ByVal **MaskName** As String)

Member of **CorelScript**

Parameters	Description
MaskName	Description of MaskName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskChannelAdd')} **Related Topics**

CorelScript.MaskChannelDelete

Sub **MaskChannelDelete**(ByVal **MaskID** As Long)

Member of **CorelScript**

Parameters	Description
MaskID	Description of MaskID goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskChannelDelete')} **Related Topics**

CorelScript.MaskChannelLoad

Sub **MaskChannelLoad**(ByVal **FileName** As String, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **LoadType** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
LoadType	Description of LoadType goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskChannelLoad')} **Related Topics**

CorelScript.MaskChannelName

Sub **MaskChannelName**(ByVal **MaskID** As Long, ByVal **Name** As String)

Member of **CorelScript**

Parameters	Description
MaskID	Description of MaskID goes here
Name	Description of Name goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskChannelName')} **Related Topics**

CorelScript.MaskChannelSave

Sub **MaskChannelSave**(ByVal **MaskID** As Long, ByVal **FileName** As String, ByVal **FilterID** As Long, ByVal **Compression** As Long)

Member of **CorelScript**

Parameters	Description
MaskID	Description of MaskID goes here
FileName	Description of FileName goes here
FilterID	Description of FilterID goes here
Compression	Description of Compression goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskChannelSave')} **Related Topics**

CorelScript.MaskChannelToMask

Sub **MaskChannelToMask**(ByVal **MaskID** As Long, ByVal **DrawMode** As Long)

Member of **CorelScript**

Parameters	Description
MaskID	Description of MaskID goes here
DrawMode	Description of DrawMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskChannelToMask')} **Related Topics**

CorelScript.MaskCreate

Sub **MaskCreate**(ByVal **PreserveImage** As Boolean, ByVal **DrawMode** As Long)

Member of **CorelScript**

Parameters	Description
PreserveImage	Description of PreserveImage goes here
DrawMode	Description of DrawMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskCreate')} **Related Topics**

CorelScript.MaskCreateFromPath

Sub **MaskCreateFromPath**(ByVal **AntiAlias** As Boolean, ByVal **DrawMode** As Long)

Member of **CorelScript**

Parameters	Description
AntiAlias	Description of AntiAlias goes here
DrawMode	Description of DrawMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskCreateFromPath')} **Related Topics**

CorelScript.MaskDeFloat

Sub **MaskDeFloat**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskDeFloat')} **Related Topics**

CorelScript.MaskDefloatIntoSelection

Sub **MaskDefloatIntoSelection**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskDefloatIntoSelection')} **Related Topics**

CorelScript.MaskDistort

Sub **MaskDistort**(ByVal **Corner1X** As Long, ByVal **Corner1Y** As Long, ByVal **Corner2X** As Long, ByVal **Corner2Y** As Long, ByVal **Corner3X** As Long, ByVal **Corner3Y** As Long, ByVal **Corner4X** As Long, ByVal **Corner4Y** As Long, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Corner1X	Description of Corner1X goes here
Corner1Y	Description of Corner1Y goes here
Corner2X	Description of Corner2X goes here
Corner2Y	Description of Corner2Y goes here
Corner3X	Description of Corner3X goes here
Corner3Y	Description of Corner3Y goes here
Corner4X	Description of Corner4X goes here
Corner4Y	Description of Corner4Y goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskDistort')} **Related Topics**

CorelScript.MaskEllipse

Sub **MaskEllipse**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **DrawMode** As Long, ByVal **Feather** As Long, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
DrawMode	Description of DrawMode goes here
Feather	Description of Feather goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskEllipse')} **Related Topics**

CorelScript.MaskExpand

Sub **MaskExpand**(ByVal **Width** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskExpand')} **Related Topics**

CorelScript.MaskFeather

Sub **MaskFeather**(ByVal **Width** As Long, ByVal **Direction** As Long, ByVal **Type** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Direction	Description of Direction goes here
Type	Description of Type goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskFeather')} **Related Topics**

CorelScript.MaskFlipHorizontal

Sub **MaskFlipHorizontal**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskFlipHorizontal')} **Related Topics**

CorelScript.MaskFlipVertical

Sub **MaskFlipVertical**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskFlipVertical')} **Related Topics**

CorelScript.MaskFloaterMoveTo

Sub **MaskFloaterMoveTo**(ByVal **Left** As Long, ByVal **Bottom** As Long, ByVal **Copy** As Boolean, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Bottom	Description of Bottom goes here
Copy	Description of Copy goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskFloaterMoveTo')} **Related Topics**

CorelScript.MaskFloaterTranslate

Sub **MaskFloaterTranslate**(ByVal **Left** As Long, ByVal **Bottom** As Long, ByVal **Copy** As Boolean)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Bottom	Description of Bottom goes here
Copy	Description of Copy goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskFloaterTranslate')} **Related Topics**

CorelScript.MaskFreehand

Sub **MaskFreehand**(ByVal **DrawMode** As Long, ByVal **Feather** As Long, ByVal **AntiAlias** As Long)

Member of **CorelScript**

Parameters	Description
DrawMode	Description of DrawMode goes here
Feather	Description of Feather goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskFreehand')} **Related Topics**

CorelScript.MaskGrow

Sub **MaskGrow**(ByVal **AntiAlias** As Boolean, ByVal **MaskVisible** As Boolean, ByVal **ToleranceMode** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
AntiAlias	Description of AntiAlias goes here
MaskVisible	Description of MaskVisible goes here
ToleranceMode	Description of ToleranceMode goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskGrow')} **Related Topics**

CorelScript.MaskInvert

Sub **MaskInvert**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskInvert')} **Related Topics**

CorelScript.MaskLasso

Sub **MaskLasso**(ByVal **DrawMode** As Long, ByVal **AntiAlias** As Boolean, ByVal **MaskVisible** As Boolean)

Member of **CorelScript**

Parameters	Description
DrawMode	Description of DrawMode goes here
AntiAlias	Description of AntiAlias goes here
MaskVisible	Description of MaskVisible goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskLasso')} **Related Topics**

CorelScript.MaskLoad

Sub **MaskLoad**(ByVal **FileName** As String, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **LoadType** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
LoadType	Description of LoadType goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskLoad')} **Related Topics**

CorelScript.MaskMagicWand

Sub **MaskMagicWand**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **DrawMode** As Long, ByVal **AntiAlias** As Boolean, ByVal **MaskVisible** As Boolean, ByVal **ToleranceMode** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
DrawMode	Description of DrawMode goes here
AntiAlias	Description of AntiAlias goes here
MaskVisible	Description of MaskVisible goes here
ToleranceMode	Description of ToleranceMode goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskMagicWand')} **Related Topics**

CorelScript.MaskPaint

Sub **MaskPaint**(ByVal **Mask** As Long)

Member of **CorelScript**

Parameters	Description
Mask	Description of Mask goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskPaint')} **Related Topics**

CorelScript.MaskRectangle

Sub **MaskRectangle**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **DrawMode** As Long, ByVal **Feather** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
DrawMode	Description of DrawMode goes here
Feather	Description of Feather goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskRectangle')} **Related Topics**

CorelScript.MaskReduce

Sub **MaskReduce**(ByVal **Width** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskReduce')} **Related Topics**

CorelScript.MaskRemove

Sub **MaskRemove**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskRemove')} **Related Topics**

CorelScript.MaskRemoveHoles

Sub **MaskRemoveHoles**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskRemoveHoles')} **Related Topics**

CorelScript.MaskRotate

Sub **MaskRotate**(ByVal **XCenter** As Long, ByVal **YCenter** As Long, ByVal **Angle** As Double, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
XCenter	Description of XCenter goes here
YCenter	Description of YCenter goes here
Angle	Description of Angle goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskRotate')} **Related Topics**

CorelScript.MaskSave

Sub **MaskSave**(ByVal **FileName** As String, ByVal **FilterID** As Long, ByVal **Compression** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
FilterID	Description of FilterID goes here
Compression	Description of Compression goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskSave')} **Related Topics**

CorelScript.MaskScissors

Sub **MaskScissors**(ByVal **DrawMode** As Long, ByVal **Feather** As Long, ByVal **AntiAlias** As Boolean, ByVal **MaskVisible** As Boolean)

Member of **CorelScript**

Parameters	Description
DrawMode	Description of DrawMode goes here
Feather	Description of Feather goes here
AntiAlias	Description of AntiAlias goes here
MaskVisible	Description of MaskVisible goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskScissors')} **Related Topics**

CorelScript.MaskSelectAll

Sub **MaskSelectAll**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskSelectAll')} **Related Topics**

CorelScript.MaskSimilar

Sub **MaskSimilar**(ByVal **AntiAlias** As Boolean, ByVal **MaskVisible** As Boolean, ByVal **ToleranceMode** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
AntiAlias	Description of AntiAlias goes here
MaskVisible	Description of MaskVisible goes here
ToleranceMode	Description of ToleranceMode goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskSimilar')} **Related Topics**

CorelScript.MaskSkew

Sub **MaskSkew**(ByVal **HorizontalAngle** As Double, ByVal **VerticalAngle** As Double, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
HorizontalAngle	Description of HorizontalAngle goes here
VerticalAngle	Description of VerticalAngle goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskSkew')} **Related Topics**

CorelScript.MaskSmooth

Sub **MaskSmooth**(ByVal **Radius** As Long)

Member of **CorelScript**

Parameters	Description
Radius	Description of Radius goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskSmooth')} **Related Topics**

CorelScript.MaskStretch

Sub **MaskStretch**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **XScale** As Double, ByVal **YScale** As Double, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
XScale	Description of XScale goes here
YScale	Description of YScale goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskStretch')} **Related Topics**

CorelScript.MaskStroke

Sub **MaskStroke**(ByVal **Mode** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskStroke')} **Related Topics**

CorelScript.MaskThreshold

Sub **MaskThreshold**(ByVal **Level** As Long)

Member of **CorelScript**

Parameters	Description
Level	Description of Level goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskThreshold')} **Related Topics**

CorelScript.MaskToMaskChannel

Sub **MaskToMaskChannel**(ByVal **MaskID** As Long)

Member of **CorelScript**

Parameters	Description
MaskID	Description of MaskID goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskToMaskChannel')} **Related Topics**

CorelScript.MaskTranslate

Sub **MaskTranslate**(ByVal **X** As Long, ByVal **Y** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MaskTranslate')} **Related Topics**

CorelScript.MovieBackOne

Sub **MovieBackOne**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieBackOne')} **Related Topics**

CorelScript.MovieCreate

Sub **MovieCreate**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieCreate')} **Related Topics**

CorelScript.MovieDeleteFrame

Sub **MovieDeleteFrame**(ByVal **FromFrame** As Long, ByVal **ToFrame** As Long)

Member of **CorelScript**

Parameters	Description
FromFrame	Description of FromFrame goes here
ToFrame	Description of ToFrame goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieDeleteFrame')} **Related Topics**

CorelScript.MovieForward

Sub **MovieForward**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieForward')} **Related Topics**

CorelScript.MovieForwardOne

Sub **MovieForwardOne**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieForwardOne')} **Related Topics**

CorelScript.MovieFrameDelay

Sub **MovieFrameDelay**(ByVal **StartFrame** As Long, ByVal **EndFrame** As Long, ByVal **Delay** As Long)

Member of **CorelScript**

Parameters	Description
StartFrame	Description of StartFrame goes here
EndFrame	Description of EndFrame goes here
Delay	Description of Delay goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieFrameDelay')} **Related Topics**

CorelScript.MovieFrameRate

Sub **MovieFrameRate**(ByVal **Frames** As Long)

Member of **CorelScript**

Parameters	Description
Frames	Description of Frames goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieFrameRate')} **Related Topics**

CorelScript.MovieGotoFrame

Sub **MovieGotoFrame**(ByVal **Frame** As Long)

Member of **CorelScript**

Parameters	Description
Frame	Description of Frame goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieGotoFrame')} **Related Topics**

CorelScript.MovieInsertFile

Sub **MovieInsertFile**(ByVal **FileName** As String, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long, ByVal **LoadType** As Long, ByVal **StartFrame** As Long, ByVal **Before** As Boolean)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here
LoadType	Description of LoadType goes here
StartFrame	Description of StartFrame goes here
Before	Description of Before goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieInsertFile')} **Related Topics**

CorelScript.MovieInsertFrame

Sub **MovieInsertFrame**(ByVal **StartFrame** As Long, ByVal **NumberOfFrames** As Long, ByVal **Before** As Boolean, ByVal **CopyCurrent** As Boolean)

Member of **CorelScript**

Parameters	Description
StartFrame	Description of StartFrame goes here
NumberOfFrames	Description of NumberOfFrames goes here
Before	Description of Before goes here
CopyCurrent	Description of CopyCurrent goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieInsertFrame')} **Related Topics**

CorelScript.MovieMoveFrame

Sub **MovieMoveFrame**(ByVal **FromFrame** As Long, ByVal **ToFrame** As Long, ByVal **MoveToFrame** As Long, ByVal **Before** As Boolean)

Member of **CorelScript**

Parameters	Description
FromFrame	Description of FromFrame goes here
ToFrame	Description of ToFrame goes here
MoveToFrame	Description of MoveToFrame goes here
Before	Description of Before goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieMoveFrame')} **Related Topics**

CorelScript.MovieRewind

Sub **MovieRewind**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieRewind')} **Related Topics**

CorelScript.MovieSelectPartial

Sub **MovieSelectPartial**(ByVal **StartFrame** As Long, ByVal **EndFrame** As Long)

Member of **CorelScript**

Parameters	Description
StartFrame	Description of StartFrame goes here
EndFrame	Description of EndFrame goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_MovieSelectPartial')} **Related Topics**

CorelScript.NibSettings

Sub **NibSettings**(ByVal **FileName** As String, ByVal **NibIndex** As Long)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
NibIndex	Description of NibIndex goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_NibSettings')} **Related Topics**

CorelScript.ObjectAddClipMask

Sub **ObjectAddClipMask**(ByVal **Value** As Long)

Member of **CorelScript**

Parameters	Description
Value	Description of Value goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectAddClipMask')} **Related Topics**

CorelScript.ObjectAddClipMaskFromMask

Sub **ObjectAddClipMaskFromMask**(ByVal **Inverted** As Boolean)

Member of **CorelScript**

Parameters	Description
Inverted	Description of Inverted goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectAddClipMaskFromMask')} **Related Topics**

CorelScript.ObjectAddClipMaskFromTransparency

Sub **ObjectAddClipMaskFromTransparency**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectAddClipMaskFromTransparency')} **Related Topics**

CorelScript.ObjectAffineDistort

Sub **ObjectAffineDistort**(ByVal **Xoffset** As Long, ByVal **Yoffset** As Long, ByVal **d11** As Double, ByVal **d12** As Double, ByVal **d21** As Double, ByVal **d22** As Double, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Xoffset	Description of Xoffset goes here
Yoffset	Description of Yoffset goes here
d11	Description of d11 goes here
d12	Description of d12 goes here
d21	Description of d21 goes here
d22	Description of d22 goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectAffineDistort')} **Related Topics**

CorelScript.ObjectAlign

Sub **ObjectAlign**(ByVal **Horizontal** As Long, ByVal **Vertical** As Long, ByVal **Center** As Boolean, ByVal **Grid** As Boolean, ByVal **Active** As Boolean, ByVal **Selected** As Boolean, ByVal **DistributeBounds** As Boolean, ByVal **HorizontalAlign** As Boolean, ByVal **VerticalAlign** As Boolean, ByVal **ObjectSpacing** As Boolean, ByVal **XSpacing** As Long, ByVal **YSpacing** As Long)

Member of **CorelScript**

Parameters	Description
Horizontal	Description of Horizontal goes here
Vertical	Description of Vertical goes here
Center	Description of Center goes here
Grid	Description of Grid goes here
Active	Description of Active goes here
Selected	Description of Selected goes here
DistributeBounds	Description of DistributeBounds goes here
HorizontalAlign	Description of HorizontalAlign goes here
VerticalAlign	Description of VerticalAlign goes here
ObjectSpacing	Description of ObjectSpacing goes here
XSpacing	Description of XSpacing goes here
YSpacing	Description of YSpacing goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectAlign')} **Related Topics**

CorelScript.ObjectClip

Sub **ObjectClip**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectClip')} **Related Topics**

CorelScript.ObjectClipToParent

Sub **ObjectClipToParent**(ByVal **ObjectID** As Long, ByVal **Clip** As Boolean)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Clip	Description of Clip goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectClipToParent')} **Related Topics**

CorelScript.ObjectColorTransparencyTool

Sub **ObjectColorTransparencyTool**(ByVal **ToleranceMode** As Long, ByVal **ApplyToClipMask** As Boolean, ByVal **Smoothing** As Long)

Member of **CorelScript**

Parameters	Description
ToleranceMode	Description of ToleranceMode goes here
ApplyToClipMask	Description of ApplyToClipMask goes here
Smoothing	Description of Smoothing goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectColorTransparencyTool')} **Related Topics**

CorelScript.ObjectCombine

Sub **ObjectCombine**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectCombine')} **Related Topics**

CorelScript.ObjectCreate

Sub **ObjectCreate**(ByVal **PreserveImage** As Boolean)

Member of **CorelScript**

Parameters	Description
PreserveImage	Description of PreserveImage goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectCreate')} **Related Topics**

CorelScript.ObjectCreateFromBackground

Sub **ObjectCreateFromBackground**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectCreateFromBackground')} **Related Topics**

CorelScript.ObjectDefringe

Sub **ObjectDefringe**(ByVal **Amount** As Long)

Member of **CorelScript**

Parameters	Description
Amount	Description of Amount goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDefringe')} **Related Topics**

CorelScript.ObjectDelete

Sub **ObjectDelete**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDelete')} **Related Topics**

CorelScript.ObjectDistort

Sub **ObjectDistort**(ByVal **Corner1X** As Long, ByVal **Corner1Y** As Long, ByVal **Corner2X** As Long, ByVal **Corner2Y** As Long, ByVal **Corner3X** As Long, ByVal **Corner3Y** As Long, ByVal **Corner4X** As Long, ByVal **Corner4Y** As Long, ByVal **AntiAlias** As Boolean)

Member of **CorelScript**

Parameters	Description
Corner1X	Description of Corner1X goes here
Corner1Y	Description of Corner1Y goes here
Corner2X	Description of Corner2X goes here
Corner2Y	Description of Corner2Y goes here
Corner3X	Description of Corner3X goes here
Corner3Y	Description of Corner3Y goes here
Corner4X	Description of Corner4X goes here
Corner4Y	Description of Corner4Y goes here
AntiAlias	Description of AntiAlias goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDistort')} **Related Topics**

CorelScript.ObjectDropShadow

Sub **ObjectDropShadow**(ByVal **Mode** As Boolean, ByVal **Direction** As Long, ByVal **Distance** As Long, ByVal **Feather** As Long, ByVal **Type** As Long, ByVal **Edges** As Long, ByVal **Opacity** As Long, ByVal **Relative** As Boolean, ByVal **Fade** As Long, ByVal **Anchor** As Long, ByVal **Stretch** As Double, ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here
Direction	Description of Direction goes here
Distance	Description of Distance goes here
Feather	Description of Feather goes here
Type	Description of Type goes here
Edges	Description of Edges goes here
Opacity	Description of Opacity goes here
Relative	Description of Relative goes here
Fade	Description of Fade goes here
Anchor	Description of Anchor goes here
Stretch	Description of Stretch goes here
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDropShadow')} **Related Topics**

CorelScript.ObjectDropShadowCombine

Sub **ObjectDropShadowCombine**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDropShadowCombine')} **Related Topics**

CorelScript.ObjectDropShadowDelete

Sub **ObjectDropShadowDelete**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDropShadowDelete')} **Related Topics**

CorelScript.ObjectDropShadowSplit

Sub **ObjectDropShadowSplit**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDropShadowSplit')} **Related Topics**

CorelScript.ObjectDuplicate

Sub **ObjectDuplicate**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectDuplicate')} **Related Topics**

CorelScript.ObjectEdit

Sub **ObjectEdit**(ByVal **ObjectID** As Long, ByVal **Edit** As Boolean)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Edit	Description of Edit goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectEdit')} **Related Topics**

CorelScript.ObjectEditAll

Sub **ObjectEditAll**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectEditAll')} **Related Topics**

CorelScript.ObjectEditNone

Sub **ObjectEditNone**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectEditNone')} **Related Topics**

CorelScript.ObjectEditSelected

Sub **ObjectEditSelected**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectEditSelected')} **Related Topics**

CorelScript.ObjectEditTransparency

Sub **ObjectEditTransparency**(ByVal **Mode** As Long)

Member of **CorelScript**

Parameters	Description
Mode	Description of Mode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectEditTransparency')} **Related Topics**

CorelScript.ObjectFeather

Sub **ObjectFeather**(ByVal **Width** As Long, ByVal **Type** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Type	Description of Type goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectFeather')} **Related Topics**

CorelScript.ObjectFlipHorizontal

Sub **ObjectFlipHorizontal**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectFlipHorizontal')} **Related Topics**

CorelScript.ObjectFlipVertical

Sub **ObjectFlipVertical**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectFlipVertical')} **Related Topics**

CorelScript.ObjectGroup

Sub **ObjectGroup**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectGroup')} **Related Topics**

CorelScript.ObjectMerge

Sub **ObjectMerge**(ByVal **All** As Boolean)

Member of **CorelScript**

Parameters	Description
All	Description of All goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectMerge')} **Related Topics**

CorelScript.ObjectMergeMode

Sub **ObjectMergeMode**(ByVal **MergeMode** As Long)

Member of **CorelScript**

Parameters	Description
MergeMode	Description of MergeMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectMergeMode')} **Related Topics**

CorelScript.ObjectName

Sub **ObjectName**(ByVal **ObjectID** As Long, ByVal **Name** As String)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Name	Description of Name goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectName')} **Related Topics**

CorelScript.ObjectNew

Sub **ObjectNew**(ByVal **Name** As String, ByVal **Opacity** As Long, ByVal **MergeMode** As Long)

Member of **CorelScript**

Parameters	Description
Name	Description of Name goes here
Opacity	Description of Opacity goes here
MergeMode	Description of MergeMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectNew')} **Related Topics**

CorelScript.ObjectOpacity

Sub **ObjectOpacity**(ByVal **Amount** As Long)

Member of **CorelScript**

Parameters	Description
Amount	Description of Amount goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectOpacity')} **Related Topics**

CorelScript.ObjectOrder

Sub **ObjectOrder**(ByVal **Order** As Long)

Member of **CorelScript**

Parameters	Description
Order	Description of Order goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectOrder')} **Related Topics**

CorelScript.ObjectOrderChange

Sub **ObjectOrderChange**(ByVal **Source** As Long, ByVal **Destination** As Long)

Member of **CorelScript**

Parameters	Description
Source	Description of Source goes here
Destination	Description of Destination goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectOrderChange')} **Related Topics**

CorelScript.ObjectProperties

Sub **ObjectProperties**(ByVal **ObjectID** As Long, ByVal **Name** As String, ByVal **Opacity** As Long, ByVal **MergeMode** As Long, ByVal **Visible** As Boolean, ByVal **Clip** As Boolean, ByVal **Enable** As Boolean, ByVal **Link** As Boolean)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Name	Description of Name goes here
Opacity	Description of Opacity goes here
MergeMode	Description of MergeMode goes here
Visible	Description of Visible goes here
Clip	Description of Clip goes here
Enable	Description of Enable goes here
Link	Description of Link goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectProperties')} **Related Topics**

CorelScript.ObjectRemoveClipMask

Sub **ObjectRemoveClipMask**(ByVal **Apply** As Boolean)

Member of **CorelScript**

Parameters	Description
Apply	Description of Apply goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectRemoveClipMask')} **Related Topics**

CorelScript.ObjectRemoveMatte

Sub **ObjectRemoveMatte**(ByVal **White** As Boolean)

Member of **CorelScript**

Parameters	Description
White	Description of White goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectRemoveMatte')} **Related Topics**

CorelScript.ObjectRotate

Sub **ObjectRotate**(ByVal **XCenter** As Long, ByVal **YCenter** As Long, ByVal **Angle** As Double, ByVal **AntiAlias** As Boolean, ByVal **Copy** As Boolean)

Member of **CorelScript**

Parameters	Description
XCenter	Description of XCenter goes here
YCenter	Description of YCenter goes here
Angle	Description of Angle goes here
AntiAlias	Description of AntiAlias goes here
Copy	Description of Copy goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectRotate')} **Related Topics**

CorelScript.ObjectSelect

Sub **ObjectSelect**(ByVal **ObjectID** As Long, ByVal **Selected** As Boolean)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Selected	Description of Selected goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectSelect')} **Related Topics**

CorelScript.ObjectSelectAll

Sub **ObjectSelectAll**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectSelectAll')} **Related Topics**

CorelScript.ObjectSelection

Sub **ObjectSelection**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectSelection')} **Related Topics**

CorelScript.ObjectSelectNone

Sub **ObjectSelectNone**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectSelectNone')} **Related Topics**

CorelScript.ObjectSkew

Sub **ObjectSkew**(ByVal **HorizontalAngle** As Double, ByVal **VerticalAngle** As Double, ByVal **AntiAlias** As Boolean, ByVal **Copy** As Boolean)

Member of **CorelScript**

Parameters	Description
HorizontalAngle	Description of HorizontalAngle goes here
VerticalAngle	Description of VerticalAngle goes here
AntiAlias	Description of AntiAlias goes here
Copy	Description of Copy goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectSkew')} **Related Topics**

CorelScript.ObjectStretch

Sub **ObjectStretch**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **XScale** As Double, ByVal **YScale** As Double, ByVal **AntiAlias** As Boolean, ByVal **Copy** As Boolean)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
XScale	Description of XScale goes here
YScale	Description of YScale goes here
AntiAlias	Description of AntiAlias goes here
Copy	Description of Copy goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectStretch')} **Related Topics**

CorelScript.ObjectTagWWWURL

Sub **ObjectTagWWWURL**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectTagWWWURL')} **Related Topics**

CorelScript.ObjectThreshold

Sub **ObjectThreshold**(ByVal **Threshold** As Long)

Member of **CorelScript**

Parameters	Description
Threshold	Description of Threshold goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectThreshold')} **Related Topics**

CorelScript.ObjectToggleClipMask

Sub **ObjectToggleClipMask**(ByVal **Clip** As Boolean)

Member of **CorelScript**

Parameters	Description
Clip	Description of Clip goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectToggleClipMask')} **Related Topics**

CorelScript.ObjectToggleLinkClipMask

Sub **ObjectToggleLinkClipMask**(ByVal **ObjectID** As Long, ByVal **Link** As Boolean)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Link	Description of Link goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectToggleLinkClipMask')} **Related Topics**

CorelScript.ObjectTranslate

Sub **ObjectTranslate**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **Copy** As Boolean)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
Copy	Description of Copy goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectTranslate')} **Related Topics**

CorelScript.ObjectTransparencyTool

Sub **ObjectTransparencyTool**(ByVal **Start** As Long, ByVal **End** As Long, ByVal **UseOriginal** As Boolean, ByVal **Handles** As Long, ByVal **ApplyToClipMask** As Boolean)

Member of **CorelScript**

Parameters	Description
Start	Description of Start goes here
End	Description of End goes here
UseOriginal	Description of UseOriginal goes here
Handles	Description of Handles goes here
ApplyToClipMask	Description of ApplyToClipMask goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectTransparencyTool')} **Related Topics**

CorelScript.ObjectUngroup

Sub **ObjectUngroup**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectUngroup')} **Related Topics**

CorelScript.ObjectURLInfo

Sub **ObjectURLInfo**(ByVal **ObjectID** As Long, ByVal **Region** As Long, ByVal **Address** As String, ByVal **Comments** As String)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here
Region	Description of Region goes here
Address	Description of Address goes here
Comments	Description of Comments goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectURLInfo')} **Related Topics**

CorelScript.ObjectVisible

Sub **ObjectVisible**(ByVal **StartIndex** As Long, ByVal **EndIndex** As Long, ByVal **Visible** As Boolean)

Member of **CorelScript**

Parameters	Description
StartIndex	Description of StartIndex goes here
EndIndex	Description of EndIndex goes here
Visible	Description of Visible goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ObjectVisible')} **Related Topics**

CorelScript.OverprintColor

Sub **OverprintColor**(ByVal **Number** As Long, ByVal **Cyan** As Long, ByVal **Magenta** As Long, ByVal **Yellow** As Long, ByVal **Black** As Long)

Member of **CorelScript**

Parameters	Description
Number	Description of Number goes here
Cyan	Description of Cyan goes here
Magenta	Description of Magenta goes here
Yellow	Description of Yellow goes here
Black	Description of Black goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_OverprintColor')} **Related Topics**

CorelScript.PaletteColor

Sub **PaletteColor**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long, ByVal **Index** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here
Index	Description of Index goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PaletteColor')} **Related Topics**

CorelScript.PathCreate

Sub **PathCreate**(ByVal **Nodes** As Long)

Member of **CorelScript**

Parameters	Description
Nodes	Description of Nodes goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathCreate')} **Related Topics**

CorelScript.PathCreateFromMask

Sub **PathCreateFromMask**(ByVal **Tightness** As Long, ByVal **Threshold** As Long)

Member of **CorelScript**

Parameters	Description
Tightness	Description of Tightness goes here
Threshold	Description of Threshold goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathCreateFromMask')} **Related Topics**

CorelScript.PathDelete

Sub **PathDelete**(ByVal **FromDisk** As Boolean)

Member of **CorelScript**

Parameters	Description
FromDisk	Description of FromDisk goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathDelete')} **Related Topics**

CorelScript.PathDuplicate

Sub **PathDuplicate**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathDuplicate')} **Related Topics**

CorelScript.PathEnd

Sub **PathEnd**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathEnd')} **Related Topics**

CorelScript.PathImportVector

Sub **PathImportVector**(ByVal **FileName** As String, ByVal **ScaleX** As Double, ByVal **ScaleY** As Double)

Member of **CorelScript**

Parameters	Description
FileName	Description of FileName goes here
ScaleX	Description of ScaleX goes here
ScaleY	Description of ScaleY goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathImportVector')} **Related Topics**

CorelScript.PathLoad

Sub **PathLoad**(ByVal **PathName** As String)

Member of **CorelScript**

Parameters	Description
PathName	Description of PathName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathLoad')} **Related Topics**

CorelScript.PathNew

Sub **PathNew**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathNew')} **Related Topics**

CorelScript.PathNode

Sub **PathNode**(ByVal **X** As Double, ByVal **Y** As Double, ByVal **Closed** As Boolean, ByVal **Continuity** As Long, ByVal **Type** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
Closed	Description of Closed goes here
Continuity	Description of Continuity goes here
Type	Description of Type goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathNode')} **Related Topics**

CorelScript.PathSave

Sub **PathSave**(ByVal **PathName** As String)

Member of **CorelScript**

Parameters	Description
PathName	Description of PathName goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathSave')} **Related Topics**

CorelScript.PathSetClippingPath

Sub **PathSetClippingPath**(ByVal **PathID** As Long)

Member of **CorelScript**

Parameters	Description
PathID	Description of PathID goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathSetClippingPath')} **Related Topics**

CorelScript.PathStroke

Sub **PathStroke**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathStroke')} **Related Topics**

CorelScript.PathVisible

Sub **PathVisible**(ByVal **Visible** As Boolean)

Member of **CorelScript**

Parameters	Description
Visible	Description of Visible goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PathVisible')} **Related Topics**

CorelScript.PolygonTool

Sub **PolygonTool**(ByVal **Width** As Long, ByVal **Transparency** As Long, ByVal **Joints** As Long, ByVal **MergeMode** As Long, ByVal **AntiAlias** As Boolean, ByVal **RenderObject** As Boolean, ByVal **Fill** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Transparency	Description of Transparency goes here
Joints	Description of Joints goes here
MergeMode	Description of MergeMode goes here
AntiAlias	Description of AntiAlias goes here
RenderObject	Description of RenderObject goes here
Fill	Description of Fill goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PolygonTool')} **Related Topics**

CorelScript.PressureSettings

Sub **PressureSettings**(ByVal **Size** As Long, ByVal **Transparency** As Long, ByVal **Softness** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long, ByVal **Texture** As Long, ByVal **Bleed** As Long, ByVal **Resaturate** As Long, ByVal **Mask** As Long, ByVal **Elongation** As Long)

Member of **CorelScript**

Parameters	Description
Size	Description of Size goes here
Transparency	Description of Transparency goes here
Softness	Description of Softness goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here
Texture	Description of Texture goes here
Bleed	Description of Bleed goes here
Resaturate	Description of Resaturate goes here
Mask	Description of Mask goes here
Elongation	Description of Elongation goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_PressureSettings')} **Related Topics**

CorelScript.RandomSeed

Sub **RandomSeed**(ByVal **Seed** As Long)

Member of **CorelScript**

Parameters	Description
Seed	Description of Seed goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_RandomSeed')} **Related Topics**

CorelScript.Rectangle

Sub **Rectangle**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_Rectangle')} **Related Topics**

CorelScript.RectangleTool

Sub **RectangleTool**(ByVal **Width** As Long, ByVal **Transparency** As Long, ByVal **Roundness** As Long, ByVal **MergeMode** As Long, ByVal **AntiAlias** As Boolean, ByVal **RenderObject** As Boolean, ByVal **Fill** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Transparency	Description of Transparency goes here
Roundness	Description of Roundness goes here
MergeMode	Description of MergeMode goes here
AntiAlias	Description of AntiAlias goes here
RenderObject	Description of RenderObject goes here
Fill	Description of Fill goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_RectangleTool')} **Related Topics**

CorelScript.RegisterObject

Function **RegisterObject**(ByVal **ObjectID** As String) As Boolean

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_RegisterObject')} **Related Topics**

CorelScript.RepeatSettings

Sub **RepeatSettings**(ByVal **Repeat** As Long, ByVal **Rotate** As Long, ByVal **RotateVar** As Long, ByVal **ColorFromImage** As Long, ByVal **Hue** As Long, ByVal **Lightness** As Long, ByVal **Saturation** As Long, ByVal **AccummulateAngle** As Long, ByVal **TangentToPath** As Long, ByVal **Scale** As Long, ByVal **ScaleVar** As Long, ByVal **StrokePath** As Long, ByVal **FromMask** As Boolean, ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Right** As Long, ByVal **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Repeat	Description of Repeat goes here
Rotate	Description of Rotate goes here
RotateVar	Description of RotateVar goes here
ColorFromImage	Description of ColorFromImage goes here
Hue	Description of Hue goes here
Lightness	Description of Lightness goes here
Saturation	Description of Saturation goes here
AccummulateAngle	Description of AccummulateAngle goes here
TangentToPath	Description of TangentToPath goes here
Scale	Description of Scale goes here
ScaleVar	Description of ScaleVar goes here
StrokePath	Description of StrokePath goes here
FromMask	Description of FromMask goes here
Left	Description of Left goes here
Top	Description of Top goes here
Right	Description of Right goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_RepeatSettings')} **Related Topics**

CorelScript.SelectionMoveTo

Sub **SelectionMoveTo**(ByVal **Left** As Long, ByVal **Bottom** As Long)

Member of **CorelScript**

Parameters	Description
Left	Description of Left goes here
Bottom	Description of Bottom goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SelectionMoveTo')} **Related Topics**

CorelScript.SelectionPlugin

Sub **SelectionPlugin**(ByVal **GroupName** As String, ByVal **EffectName** As String, ByVal **MemoryType** As Long, ByVal **MemorySize** As Long, ByVal **Parameters** As String)

Member of **CorelScript**

Parameters	Description
GroupName	Description of GroupName goes here
EffectName	Description of EffectName goes here
MemoryType	Description of MemoryType goes here
MemorySize	Description of MemorySize goes here
Parameters	Description of Parameters goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SelectionPlugin')} **Related Topics**

CorelScript.SetActiveTool

Sub **SetActiveTool**(ByVal **ToolID** As Long)

Member of **CorelScript**

Parameters	Description
ToolID	Description of ToolID goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SetActiveTool')} **Related Topics**

CorelScript.SetDocumentInfo

Sub **SetDocumentInfo**(ByVal **Width** As Long, ByVal **Height** As Long)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Height	Description of Height goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SetDocumentInfo')} **Related Topics**

CorelScript.SetDocVisible

Sub **SetDocVisible**(ByVal **Show** As Boolean)

Member of **CorelScript**

Parameters	Description
Show	Description of Show goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SetDocVisible')} **Related Topics**

CorelScript.SetPaintColor

Sub **SetPaintColor**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SetPaintColor')} **Related Topics**

CorelScript.SetPaperColor

Sub **SetPaperColor**(ByVal **ColorModel** As Long, ByVal **Color1** As Long, ByVal **Color2** As Long, ByVal **Color3** As Long, ByVal **Color4** As Long)

Member of **CorelScript**

Parameters	Description
ColorModel	Description of ColorModel goes here
Color1	Description of Color1 goes here
Color2	Description of Color2 goes here
Color3	Description of Color3 goes here
Color4	Description of Color4 goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SetPaperColor')} **Related Topics**

CorelScript.SetVisible

Sub **SetVisible**(ByVal **Show** As Boolean)

Member of **CorelScript**

Parameters	Description
Show	Description of Show goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SetVisible')} **Related Topics**

CorelScript.SnapToGrid

Sub **SnapToGrid**(ByVal **SnapOn** As Boolean)

Member of **CorelScript**

Parameters	Description
SnapOn	Description of SnapOn goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SnapToGrid')} **Related Topics**

CorelScript.StartCloneDraw

Sub **StartCloneDraw**(ByVal **SrcPtX** As Long, ByVal **SrcPtY** As Long, ByVal **DestPtX** As Long, ByVal **DestPtY** As Long, ByVal **Timer** As Long, ByVal **Pressure** As Long, ByVal **Tilt** As Long, ByVal **Rotate** As Long)

Member of **CorelScript**

Parameters	Description
SrcPtX	Description of SrcPtX goes here
SrcPtY	Description of SrcPtY goes here
DestPtX	Description of DestPtX goes here
DestPtY	Description of DestPtY goes here
Timer	Description of Timer goes here
Pressure	Description of Pressure goes here
Tilt	Description of Tilt goes here
Rotate	Description of Rotate goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_StartCloneDraw')} **Related Topics**

CorelScript.StartDraw

Sub **StartDraw**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **Timer** As Long, ByVal **Pressure** As Long, ByVal **Tilt** As Long, ByVal **Rotate** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
Timer	Description of Timer goes here
Pressure	Description of Pressure goes here
Tilt	Description of Tilt goes here
Rotate	Description of Rotate goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_StartDraw')} **Related Topics**

CorelScript.SymmetrySettings

Sub **SymmetrySettings**(ByVal **CenterX** As Long, ByVal **CenterY** As Long, ByVal **Mode** As Long, ByVal **NumPoints** As Long)

Member of **CorelScript**

Parameters	Description
CenterX	Description of CenterX goes here
CenterY	Description of CenterY goes here
Mode	Description of Mode goes here
NumPoints	Description of NumPoints goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_SymmetrySettings')} **Related Topics**

CorelScript.TextAppend

Sub **TextAppend**(ByVal **Text** As String)

Member of **CorelScript**

Parameters	Description
Text	Description of Text goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TextAppend')} **Related Topics**

CorelScript.TextEdit

Sub **TextEdit**(ByVal **ObjectID** As Long)

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TextEdit')} **Related Topics**

CorelScript.TextFitToPath

Sub **TextFitToPath**(ByVal **Position** As Long)

Member of **CorelScript**

Parameters	Description
Position	Description of Position goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TextFitToPath')} **Related Topics**

CorelScript.TextRender

Sub **TextRender**()

Member of **CorelScript**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TextRender')} **Related Topics**

CorelScript.TextSetting

Sub **TextSetting**(ByVal **Arg** As String, ByVal **Value** As String)

Member of **CorelScript**

Parameters	Description
Arg	Description of Arg goes here
Value	Description of Value goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TextSetting')} **Related Topics**

CorelScript.TextTool

Sub **TextTool**(ByVal **X** As Long, ByVal **Y** As Long, ByVal **RenderToMask** As Boolean, ByVal **AntiAlias** As Boolean, ByVal **DrawMode** As Long)

Member of **CorelScript**

Parameters	Description
X	Description of X goes here
Y	Description of Y goes here
RenderToMask	Description of RenderToMask goes here
AntiAlias	Description of AntiAlias goes here
DrawMode	Description of DrawMode goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TextTool')} **Related Topics**

CorelScript.ToleranceSettings

Sub **ToleranceSettings**(ByVal **ToleranceMode** As Long, ByVal **Normal** As Long, ByVal **Hue** As Long, ByVal **Saturation** As Long, ByVal **Brightness** As Long)

Member of **CorelScript**

Parameters	Description
ToleranceMode	Description of ToleranceMode goes here
Normal	Description of Normal goes here
Hue	Description of Hue goes here
Saturation	Description of Saturation goes here
Brightness	Description of Brightness goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_ToleranceSettings')} **Related Topics**

CorelScript.TransparencyBrushTool

Sub **TransparencyBrushTool**(ByVal **Width** As Long, ByVal **Flatten** As Long, ByVal **Rotate** As Long, ByVal **NibShape** As Long, ByVal **Transparency** As Long, ByVal **SoftEdge** As Long, ByVal **Opacity** As Long, ByVal **UseOriginal** As Boolean, ByVal **AntiAlias** As Boolean, ByVal **ApplyToClipMask** As Boolean)

Member of **CorelScript**

Parameters	Description
Width	Description of Width goes here
Flatten	Description of Flatten goes here
Rotate	Description of Rotate goes here
NibShape	Description of NibShape goes here
Transparency	Description of Transparency goes here
SoftEdge	Description of SoftEdge goes here
Opacity	Description of Opacity goes here
UseOriginal	Description of UseOriginal goes here
AntiAlias	Description of AntiAlias goes here
ApplyToClipMask	Description of ApplyToClipMask goes here

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_TransparencyBrushTool')} **Related Topics**

CorelScript.UnRegisterObject

Function **UnRegisterObject**(ByVal **ObjectID** As String) As Boolean

Member of **CorelScript**

Parameters	Description
ObjectID	Description of ObjectID goes here

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScript;FNC_UnRegisterObject')} **Related Topics**

CorelScriptFile properties

CorelScriptFile Legend

- Application

- FileName
  Name

- Parent

CorelScriptFile methods

CorelScriptFile Legend

- Delete
- Play
- Translate

CorelScriptFile

Class **CorelScriptFile**

Properties Methods Referenced By

The **CorelScriptFile** class defines the characteristics of CorelSCRIPT file objects and describes the look and behavior of the objects through its properties and methods. Corel SCRIPT files are individual packages of Corel SCRIPT programming procedures that perform a specific task when used within the Corel PHOTO-PAINT application.

Corel SCRIPT is an application included with some Corel applications to record, play, and run recordings and scripts. Corel SCRIPT starts automatically when you record or play a recording from a Corel application that supports scripts. If you run a script for an application that is not running, Corel SCRIPT automatically starts the application.

A typical user can take advantage of Corel SCRIPT to automate repetitive tasks with scripts written in the Corel SCRIPT programming language. However, Corel SCRIPT is more than just a language that you can use to create and run application script files; it's also a powerful programming language that can be used as a standalone application.

The Corel SCRIPT programming language is based on the BASIC programming language. If you're already familiar with a version of BASIC, you'll find the Corel SCRIPT programming language easy to read and understand.

For more information on Corel SCRIPT and its components, please view the Help File in the Corel SCRIPT Editor.

{button ,AL(`CLS_CorelScriptFile')} **Related Topics**

CorelScriptFile.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_CorelScriptFile;FNC_Application')} **Related Topics**

CorelScriptFile.FileName

Property **FileName** As String

Description

The **FileName** property returns a string value that identifies the full computer location and file name of a Corel SCRIPT file in Corel PHOTO-PAINT. Corel SCRIPT files are individual packages of Corel SCRIPT programming procedures that perform a specific task when used within the Corel PHOTO-PAINT application.

Corel SCRIPT is an application included with some Corel applications to record, play, and run recordings and scripts. Corel SCRIPT starts automatically when you record or play a recording from a Corel application that supports scripts. A typical user can take advantage of Corel SCRIPT to automate repetitive tasks with scripts written in the Corel SCRIPT programming language. However, Corel SCRIPT is more than just a language that you can use to create and run application script files; it's also a powerful programming language that can be used as a standalone application.

The **FileName** property returns a Read-Only value.

{button ,AL(`CLS_CorelScriptFile;FNC_FileName')} **Related Topics**

CorelScriptFile.Name

Property **Name** As String

Description

The **Name** property returns or sets a string value that identifies a new VBA code module, when it is translated from a Corel SCRIPT file. Corel SCRIPT files are individual packages of Corel SCRIPT programming procedures that perform a specific task when used within the Corel PHOTO-PAINT application.

Corel SCRIPT is an application included with some Corel applications to record, play, and run recordings and scripts. Corel SCRIPT starts automatically when you record or play a recording from a Corel application that supports scripts. A typical user can take advantage of Corel SCRIPT to automate repetitive tasks with scripts written in the Corel SCRIPT programming language. However, Corel SCRIPT is more than just a language that you can use to create and run application script files; it's also a powerful programming language that can be used as a standalone application.

A VBA code module is a programming package of Visual Basic for Applications code segments. VBA is a subset of the Microsoft Visual Basic (VB) object-oriented programming environment. VBA uses the Visual Basic Editor interactive development environment and the VB programming language to enhance applications by manipulating the application's objects, exposed by its object model. VBA can access other applications from within Corel PHOTO-PAINT 10.0 by referencing that application's object model components.

By default, the module name is set to the original SCRIPT name, without the script extension (.csc).

{button ,AL(`CLS_CorelScriptFile;FNC_Name')} **Related Topics**

CorelScriptFile.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_CorelScriptFile;FNC_Parent')} **Related Topics**

CorelScriptFile.Delete

Sub **Delete**()

Description

The **Delete** method removes the original Corel SCRIPT file from Corel PHOTO-PAINT.Corel SCRIPT files are individual packages of Corel SCRIPT programming procedures that perform a specific task when used within the Corel PHOTO-PAINT application.

Corel SCRIPT is an application included with some Corel applications to record, play, and run recordings and scripts. Corel SCRIPT starts automatically when you record or play a recording from a Corel application that supports scripts. A typical user can take advantage of Corel SCRIPT to automate repetitive tasks with scripts written in the Corel SCRIPT programming language. However, Corel SCRIPT is more than just a language that you can use to create and run application script files; it's also a powerful programming language that can be used as a standalone application.

{button ,AL(`CLS_CorelScriptFile;FNC_Delete')} **Related Topics**

CorelScriptFile.Play

Sub **Play**([ByVal Document As Object = Nothing])

Description

The **Play** method runs a Corel SCRIPT file in the active document of Corel PHOTO-PAINT. Corel SCRIPT files are individual packages of Corel SCRIPT programming procedures that perform a specific task when used within the Corel PHOTO-PAINT application.

Corel SCRIPT is an application included with some Corel applications to record, play, and run recordings and scripts. Corel SCRIPT starts automatically when you record or play a recording from a Corel application that supports scripts. A typical user can take advantage of Corel SCRIPT to automate repetitive tasks with scripts written in the Corel SCRIPT programming language. However, Corel SCRIPT is more than just a language that you can use to create and run application script files; it's also a powerful programming language that can be used as a standalone application.

In order to use the **Play** method, you must pass the **Document** value as a parameter:

Parameters	Description
Document	The **Document** parameter specifies which document will run the script file and activates that document prior to running the script. If a document is not specified with the **Play** method, the active document with run the script.

{button ,AL(`CLS_CorelScriptFile;FNC_Play')} **Related Topics**

CorelScriptFile.Translate

Sub **Translate**([ByVal Document As Object = Nothing])

Description

The **Translate** method converts a Corel SCRIPT file into a VBA code module in Corel PHOTO-PAINT. Corel SCRIPT files are individual packages of Corel SCRIPT programming procedures that perform a specific task when used within the Corel PHOTO-PAINT application.

Corel SCRIPT is an application included with some Corel applications to record, play, and run recordings and scripts. Corel SCRIPT starts automatically when you record or play a recording from a Corel application that supports scripts. A typical user can take advantage of Corel SCRIPT to automate repetitive tasks with scripts written in the Corel SCRIPT programming language. However, Corel SCRIPT is more than just a language that you can use to create and run application script files; it's also a powerful programming language that can be used as a standalone application.

A VBA code module is a programming package of Visual Basic for Applications code segments. VBA is a subset of the Microsoft Visual Basic (VB) object-oriented programming environment. VBA uses the Visual Basic Editor interactive development environment and the VB programming language to enhance applications by manipulating the application's objects, exposed by its object model. VBA can access other applications from within Corel PHOTO-PAINT 10.0 by referencing that application's object model components.

In order to use the **Translate** method, you must pass the **Document** value as a parameter:

Parameters	Description
Document	The **Document** parameter specifies which document will contain the generated VBA code module. If a document is not specified with the Translate method, the generated VBA code module will be stored in Corel PHOTO-PAINT 10.0 Global Macros.

{button ,AL(`CLS_CorelScriptFile;FNC_Translate')} **Related Topics**

CorelScriptTools methods

CorelScriptTools Legend

- AngleConvert
- ASin
- BeginWaitCursor
- BuildDate
- BuildTime
- Dec
- EndWaitCursor
- FileAttr
- FindFirstFolder
- FindNextFolder
- FormatTime
- FromCentimeters
- FromCiceros
- FromDidots
- FromInches
- FromPicas
- FromPoints
- GetAppHandle
- GetColor
- GetCommandLine
- GetCurrFolder
- GetDateInfo
- GetFileBox
- GetFolder
- GetFont
- GetProcessInfo
- GetScriptFolder
- GetTempFolder
- GetTimeInfo
- GetType
- GetVersion
- GetWinHandle
- Kill
- LengthConvert
- Log
- MkFolder
- RegistryQuery
- Rename
- RmFolder
- ToCentimeters
- ToCiceros
- ToDidots
- ToInches
- ToPicas
- ToPoints

CorelScriptTools

Class **CorelScriptTools**

Methods Referenced By

CorelScriptTools Class

{button ,AL(`CLS_CorelScriptTools')} **Related Topics**

CorelScriptTools.AngleConvert

Function **AngleConvert**(ByVal **FromUnit** As Long, ByVal **ToUnit** As Long, ByVal **Value** As Double) As Double

Angle Convert

Member of **CorelScriptTools**

Parameters	Description
FromUnit	Description of FromUnit goes here (**in**)
ToUnit	Description of ToUnit goes here (**in**)
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_AngleConvert')} **Related Topics**

CorelScriptTools.ASin

Function **ASin**(ByVal **Value** As Double) As Double

Arc Sine

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ASin')} **Related Topics**

CorelScriptTools.BeginWaitCursor

Sub **BeginWaitCursor**()

Begin Wait Cursor

Member of **CorelScriptTools**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_BeginWaitCursor')} **Related Topics**

CorelScriptTools.BuildDate

Function **BuildDate**(ByVal **Year** As Long, ByVal **Month** As Long, ByVal **Day** As Long) As Date

Build Date

Member of **CorelScriptTools**

Parameters	Description
Year	Description of Year goes here (**in**)
Month	Description of Month goes here (**in**)
Day	Description of Day goes here (**in**)

Return value

Returns Date

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_BuildDate')} **Related Topics**

CorelScriptTools.BuildTime

Function **BuildTime**(ByVal **Hour** As Long, ByVal **Minute** As Long, ByVal **Second** As Long) As Date

Build Time

Member of **CorelScriptTools**

Parameters	Description
Hour	Description of Hour goes here (**in**)
Minute	Description of Minute goes here (**in**)
Second	Description of Second goes here (**in**)

Return value

Returns Date

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_BuildTime')} **Related Topics**

CorelScriptTools.Dec

Function **Dec**(ByVal **Hex** As String) As Long

Decimal

Member of **CorelScriptTools**

Parameters	Description
Hex	Description of Hex goes here (**in**)

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_Dec')} **Related Topics**

CorelScriptTools.EndWaitCursor

Sub **EndWaitCursor**()

End Wait Cursor

Member of **CorelScriptTools**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_EndWaitCursor')} **Related Topics**

CorelScriptTools.FileAttr

Function **FileAttr**(ByVal **FolderFile** As String) As Long

File Attributes

Member of **CorelScriptTools**

Parameters	Description
FolderFile	Description of FolderFile goes here (**in**)

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FileAttr')} **Related Topics**

CorelScriptTools.FindFirstFolder

Function **FindFirstFolder**(ByVal **SearchCriteria** As String, ByVal **Attributes** As Long) As String

Find First Folder

Member of **CorelScriptTools**

Parameters	Description
SearchCriteria	Description of SearchCriteria goes here (**in**)
Attributes	Description of Attributes goes here (**in**)

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FindFirstFolder')} **Related Topics**

CorelScriptTools.FindNextFolder

Function **FindNextFolder**() As String

Find Next Folder

Member of **CorelScriptTools**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FindNextFolder')} **Related Topics**

CorelScriptTools.FormatTime

Function **FormatTime**(ByVal **Time** As Date, ByVal **Format** As String) As String

Format Time

Member of **CorelScriptTools**

Parameters	Description
Time	Description of Time goes here (**in**)
Format	Description of Format goes here (**in**)

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FormatTime')} **Related Topics**

CorelScriptTools.FromCentimeters

Function **FromCentimeters**(ByVal **Value** As Double) As Double

From Centimeters

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FromCentimeters')} **Related Topics**

CorelScriptTools.FromCiceros

Function **FromCiceros**(ByVal **Value** As Double) As Double

From Ciceros

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FromCiceros')} **Related Topics**

CorelScriptTools.FromDidots

Function **FromDidots**(ByVal **Value** As Double) As Double

From Didots

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FromDidots')} **Related Topics**

CorelScriptTools.FromInches

Function **FromInches**(ByVal **Value** As Double) As Double

From Inches

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FromInches')} **Related Topics**

CorelScriptTools.FromPicas

Function **FromPicas**(ByVal **Value** As Double) As Double

From Picas

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FromPicas')} **Related Topics**

CorelScriptTools.FromPoints

Function **FromPoints**(ByVal **Value** As Double) As Double

From Points

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_FromPoints')} **Related Topics**

CorelScriptTools.GetAppHandle

Function **GetAppHandle**() As Long

Gets Application Handle

Member of **CorelScriptTools**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetAppHandle')} **Related Topics**

CorelScriptTools.GetColor

Function **GetColor**(ByRef **Red** As Long, ByRef **Green** As Long, ByRef **Blue** As Long) As Boolean

Gets Color

Member of **CorelScriptTools**

Parameters	Description
Red	Description of Red goes here (**in**,**out**)
Green	Description of Green goes here (**in**,**out**)
Blue	Description of Blue goes here (**in**,**out**)

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetColor')} **Related Topics**

CorelScriptTools.GetCommandLine

Function **GetCommandLine**() As String

Gets Command Line

Member of **CorelScriptTools**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetCommandLine')} **Related Topics**

CorelScriptTools.GetCurrFolder

Function **GetCurrFolder**() As String

Gets Current Folder

Member of **CorelScriptTools**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetCurrFolder')} **Related Topics**

CorelScriptTools.GetDateInfo

Sub **GetDateInfo**(ByVal **Date** As Date, ByRef **Year** As Long, ByRef **Month** As Long, ByRef **Day** As Long, ByRef **DayOfWeek** As Long)

Gets Date Info

Member of **CorelScriptTools**

Parameters	Description
Date	Description of Date goes here (**in**)
Year	Description of Year goes here (**out**)
Month	Description of Month goes here (**out**)
Day	Description of Day goes here (**out**)
DayOfWeek	Description of DayOfWeek goes here (**out**)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetDateInfo')} **Related Topics**

CorelScriptTools.GetFileBox

Function **GetFileBox**([ByVal Filter As String = "All Files (*.*)|*.*"], [ByVal Title As String = ""], [ByVal Type As Long = 0], [ByVal File As String = ""], [ByVal Extension As String = ""], [ByVal Folder As String = ""], [ByVal Button As String = ""]) As String

Gets File Box

Member of **CorelScriptTools**

Parameters	Description
Filter	Description of Filter goes here (**in**) Optional Default value = **"All Files (*.*)|*.*"**
Title	Description of Title goes here (**in**) Optional Default value = **""**
Type	Description of Type goes here (**in**) Optional Default value = **0**
File	Description of File goes here (**in**) Optional Default value = **""**
Extension	Description of Extension goes here (**in**) Optional Default value = **""**
Folder	Description of Folder goes here (**in**) Optional Default value = **""**
Button	Description of Button goes here (**in**) Optional Default value = **""**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetFileBox')} **Related Topics**

CorelScriptTools.GetFolder

Function **GetFolder**([ByVal InitFolder As String = ""]) As String

Gets Folder

Member of **CorelScriptTools**

Parameters	Description
InitFolder	Description of InitFolder goes here (**in**) Optional Default value = **""**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetFolder')} **Related Topics**

CorelScriptTools.GetFont

Function **GetFont**(ByRef **FaceName** As String, ByRef **PointSize** As Long, ByRef **Weight** As Long, ByRef **Italic** As Boolean, ByRef **Underline** As Boolean, ByRef **StrikeOut** As Boolean, ByRef **Red** As Long, ByRef **Green** As Long, ByRef **Blue** As Long) As Boolean

Gets Font

Member of **CorelScriptTools**

Parameters	Description
FaceName	Description of FaceName goes here (**in**,**out**)
PointSize	Description of PointSize goes here (**in**,**out**)
Weight	Description of Weight goes here (**in**,**out**)
Italic	Description of Italic goes here (**in**,**out**)
Underline	Description of Underline goes here (**in**,**out**)
StrikeOut	Description of StrikeOut goes here (**in**,**out**)
Red	Description of Red goes here (**in**,**out**)
Green	Description of Green goes here (**in**,**out**)
Blue	Description of Blue goes here (**in**,**out**)

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetFont')} **Related Topics**

CorelScriptTools.GetProcessInfo

Function **GetProcessInfo**(ByVal **ProcessHandle** As Long) As Long

Gets Process Info

Member of **CorelScriptTools**

Parameters	Description
ProcessHandle	Description of ProcessHandle goes here (**in**)

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetProcessInfo')} **Related Topics**

CorelScriptTools.GetScriptFolder

Function **GetScriptFolder**() As String

Gets Script Folder

Member of **CorelScriptTools**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetScriptFolder')} **Related Topics**

CorelScriptTools.GetTempFolder

Function **GetTempFolder**() As String

Gets Temp Folder

Member of **CorelScriptTools**

Return value

Returns String

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetTempFolder')} **Related Topics**

CorelScriptTools.GetTimeInfo

Sub **GetTimeInfo**(ByVal **Time** As Date, ByRef **Hour** As Long, ByRef **Minute** As Long, ByRef **Second** As Long)

Gets Time Info

Member of **CorelScriptTools**

Parameters	Description
Time	Description of Time goes here (**in**)
Hour	Description of Hour goes here (**out**)
Minute	Description of Minute goes here (**out**)
Second	Description of Second goes here (**out**)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetTimeInfo')} **Related Topics**

CorelScriptTools.GetType

Function **GetType**(ByVal **Expression** As Variant) As Long

Gets Type

Member of **CorelScriptTools**

Parameters	Description
Expression	Description of Expression goes here (**in**)

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetType')} **Related Topics**

CorelScriptTools.GetVersion

Function **GetVersion**(ByVal **Option** As Long) As Long

Gets Version

Member of **CorelScriptTools**

Parameters	Description
Option	Description of Option goes here (**in**)

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetVersion')} **Related Topics**

CorelScriptTools.GetWinHandle

Function **GetWinHandle**() As Long

Gets Window Handle

Member of **CorelScriptTools**

Return value

Returns Long

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_GetWinHandle')} **Related Topics**

CorelScriptTools.Kill

Function **Kill**(ByVal **FileName** As String) As Boolean

Kill

Member of **CorelScriptTools**

Parameters	Description
FileName	Description of FileName goes here (**in**)

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_Kill')} **Related Topics**

CorelScriptTools.LengthConvert

Function **LengthConvert**(ByVal **FromUnit** As Long, ByVal **ToUnit** As Long, ByVal **Value** As Double) As Double

Length Convert

Member of **CorelScriptTools**

Parameters	Description
FromUnit	Description of FromUnit goes here (**in**)
ToUnit	Description of ToUnit goes here (**in**)
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_LengthConvert')} **Related Topics**

CorelScriptTools.Log

Function **Log**(ByVal **Value** As Double) As Double

Log

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_Log')} **Related Topics**

CorelScriptTools.MkFolder

Function **MkFolder**(ByVal **Folder** As String) As Boolean

Makes Folder

Member of **CorelScriptTools**

Parameters	Description
Folder	Description of Folder goes here (**in**)

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_MkFolder')} **Related Topics**

CorelScriptTools.RegistryQuery

Function **RegistryQuery**(ByVal **MainKey** As Long, ByVal **SubKey** As String, ByVal **Value** As String) As Variant

Registry Query

Member of **CorelScriptTools**

Parameters	Description
MainKey	Description of MainKey goes here (**in**)
SubKey	Description of SubKey goes here (**in**)
Value	Description of Value goes here (**in**)

Return value

Returns Variant

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_RegistryQuery')} **Related Topics**

CorelScriptTools.Rename

Function **Rename**(ByVal **Src** As String, ByVal **Dst** As String, [ByVal Overwrite As Long = 1]) As Boolean

Rename

Member of **CorelScriptTools**

Parameters	Description
Src	Description of Src goes here (**in**)
Dst	Description of Dst goes here (**in**)
Overwrite	Description of Overwrite goes here (**in**) Optional Default value = **1**

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_Rename')} **Related Topics**

CorelScriptTools.RmFolder

Function **RmFolder**(ByVal **Folder** As String) As Boolean

Removes Folder

Member of **CorelScriptTools**

Parameters	Description
Folder	Description of Folder goes here (**in**)

Return value

Returns Boolean

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_RmFolder')} **Related Topics**

CorelScriptTools.ToCentimeters

Function **ToCentimeters**(ByVal **Value** As Double) As Double

To Centimeters

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ToCentimeters')} **Related Topics**

CorelScriptTools.ToCiceros

Function **ToCiceros**(ByVal **Value** As Double) As Double

To Ciceros

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ToCiceros')} **Related Topics**

CorelScriptTools.ToDidots

Function **ToDidots**(ByVal **Value** As Double) As Double

To Didots

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ToDidots')} **Related Topics**

CorelScriptTools.ToInches

Function **ToInches**(ByVal **Value** As Double) As Double

To Inches

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ToInches')} **Related Topics**

CorelScriptTools.ToPicas

Function **ToPicas**(ByVal **Value** As Double) As Double

To Picas

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ToPicas')} **Related Topics**

CorelScriptTools.ToPoints

Function **ToPoints**(ByVal **Value** As Double) As Double

To Points

Member of **CorelScriptTools**

Parameters	Description
Value	Description of Value goes here (**in**)

Return value

Returns Double

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_CorelScriptTools;FNC_ToPoints')} **Related Topics**

Document properties

Document Legend

- ActiveFrame
- ActiveLayer
- ActiveWindow
- Application
- Background
- Channels
- ColorTable
  - Dirty
  - DpiX
  - DpiY
- FileName
- FilePath
- Frames
- FullFileName
- Guides
- IsMovie
- Layers
  - LockTransparency
- Mask
- Mode
- MovieFrameCount
- MovieStartFrame
  - Name
- Parent
- SizeHeight
- SizeWidth
- Tools
  - VirtualHeight
  - VirtualWidth
- Windows

Document methods

Document Legend

- Activate
- AddBackground
- ApplyICCProfile
- Checkpoint
- Close
- ConvertTo
- ConvertToBW
- ConvertToDuotone
- ConvertToMovie
- ConvertToNTSC
- ConvertToPAL
- ConvertToPaletted
- CreateGuide
- CreatePalette
- CreatePaletteFromVisible
- Crop
- CropToColor
- CropToMask
- DeskewCrop
- Duplicate
- Flip
- PaperSize
- PrintOut
- Redo
- Repeat
- Resample
- RestoreCheckpoint
- Revert
- Rotate
- Save
- SaveAs
- SelectMovieFrames
- SetResolution
- Split
- Undo

Document

Class **Document**

Properties Methods Referenced By

The **Document** class defines the characteristics of document objects and describes the look and behavior of the collection objects through its properties and methods. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

New documents use the preset options built into the Corel PHOTO-PAINT application. Every new document allows you to save settings so that new images opened in Corel PHOTO-PAINT open to the settings established in this Document page. These options include:

- Options set in the Options dialog box
- Toolbar settings
- Docker window
- Color palette

You can add, delete, and rename pages in documents as well as create web documents. Detailed information about each Corel PHOTO-PAINT document can be viewed and saved in a text file within a word processing application for future use.

{button ,AL(`CLS_Document')} **Related Topics**

Document.ActiveFrame

Property **ActiveFrame** As **Frame**

Description

The **ActiveFrame** property returns a value associated with the active frame of a document in Corel PHOTO-PAINT.A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

The **ActiveFrame** property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_ActiveFrame')} **Related Topics**

Document.ActiveLayer

Property **ActiveLayer** As **Layer**

Description

The **ActiveLayer** property returns a value associated with the active layer of a document in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

The **ActiveLayer** property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_ActiveLayer')} **Related Topics**

Document.ActiveWindow

Property **ActiveWindow** As **Window**

Description

The **ActiveWindow** property returns a value associated with the active window of a document in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

The **ActiveWindow** property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_ActiveWindow')} **Related Topics**

Document.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Application')} **Related Topics**

Document.Background

Property **Background** As **Background**

Description

The **Background** property returns a value associated with the image background of a document in Corel PHOTO-PAINT. When you create an image, you specify the color mode and the background color you want to use. The color mode defines the number of colors in a bitmap image. For example, if you choose the Black-and-White color mode, there are two colors in the image (black and white). By default, the background is white; however, you can choose another color.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Background')} **Related Topics**

Document.Channels

Property **Channels** As **Channels**

Description

The **Channels** property returns a value associated with the Channels collection in a document in Corel PHOTO-PAINT. A channel is an 8-bit grayscale image that stores color or mask information for another image. There are two types of channels: color and mask. Images have one color channel for each component of the color model on which they are based. Each channel contains the color information for that component. Mask (alpha) channels store masks that you create for your images and are saved with images in formats that support mask information (e.g., .CPT).

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Channels')} **Related Topics**

Document.ColorTable

Property **ColorTable** As **ColorTable**

Description

The **ColorTable** property returns a value associated with a color table in a document in Corel PHOTO-PAINT. A color table is a feature in Corel PHOTO-PAINT that edits colors in a paletted image. A color table lets you edit a color in a paletted color image.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_ColorTable')} **Related Topics**

Document.Dirty

Property **Dirty** As Boolean

Description

The **Dirty** property returns or sets a True or False value that allows users to determine if the document is dirty and if so, set the dirty flag in the document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property determines whether the document was modified after it was last saved.

{button ,AL(`CLS_Document;FNC_Dirty')} **Related Topics**

Document.DpiX

Property **DpiX** As Long

Description

The **DpiX** property returns or set the horizontal resolution of the active image, in dots per inch (dpi) in a document in Corel PHOTO-PAINT. Resolution refers to the amount of detail and information that an image file contains, as well as the level of detail that an input, output or display device is capable of producing. When you work with bitmaps, resolution affects the quality of your final output and the file size.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_DpiX')} **Related Topics**

Document.DpiY

Property **DpiY** As Long

Description

The **DpiY** property returns or set the vertical resolution of the active image, in dots per inch (dpi) in a document in Corel PHOTO-PAINT. Resolution refers to the amount of detail and information that an image file contains, as well as the level of detail that an input, output or display device is capable of producing. When you work with bitmaps, resolution affects the quality of your final output and the file size.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_DpiY')} **Related Topics**

Document.FileName

Property **FileName** As String

Description

The **FileName** property returns a string value associated with the name of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This value returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_FileName')} **Related Topics**

Document.FilePath

Property **FilePath** As String

Description

The **FilePath** property returns a string value associated with the computer location of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This value returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_FilePath')} **Related Topics**

Document.Frames

Property **Frames** As **Frames**

Description

The **Frames** property returns a value associated with the **Frames** collection in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Frames')} **Related Topics**

Document.FullFileName

Property **FullFileName** As String

Description

The **FullFileName** property returns a string value associated with the file path and file name of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This value returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_FullFileName')} **Related Topics**

Document.Guides

Property **Guides** As **Guides**

Description

The **Guides** property returns a value associated with the **Guides** collection of a document in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Guides')} **Related Topics**

Document.IsMovie

Property **IsMovie** As Boolean

Description

The **IsMove** property returns a True or False value that indicates whether or not a document is a move in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns Read-Only value.

{button ,AL(`CLS_Document;FNC_IsMovie')} **Related Topics**

Document.Layers

Property **Layers** As **Layers**

Description

The **Layers** property returns a value associated with the **Layers** collection of a document in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in a drawing. You can control how objects in your drawing overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex drawing.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Layers')} **Related Topics**

Document.LockTransparency

Property **LockTransparency** As Boolean

Description

The **LockTransparency** property returns or sets a True or False value that indicates whether or a transparency in a document is locked in Corel PHOTO-PAINT. Transparency is the ability to see through an item. The opposite of transparent is opaque. Setting lower levels of transparency causes higher levels of opacity and less visibility of the underlying items or image.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_LockTransparency')} **Related Topics**

Document.Mask

Property **Mask** As **Mask**

Description

The **Mask** property returns a value associated with a mask object in a document in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Mask')} **Related Topics**

Document.Mode

Property **Mode** As **cdrImageMode**

Description

The **Mode** property returns the image mode in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns cdrImageMode and is a Read-Only value.

{button ,AL(`CLS_Document;FNC_Mode')} **Related Topics**

Document.MovieFrameCount

Property **MovieFrameCount** As Long

Description

The **MovieFrameCount** returns the total number of movie frames in a document in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_MovieFrameCount')} **Related Topics**

Document.MovieStartFrame

Property **MovieStartFrame** As Long

Description

The **MovieStartFrame** returns the index number of the first frame in a document in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_MovieStartFrame')} **Related Topics**

Document.Name

Property **Name** As String

Description

The **Name** property returns or sets the name of the current document in Corel PHOTO-PAINT. The name of the current document appears in the title bar of the main application window after the application name.

The **Name** property returns a String value.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Name')} **Related Topics**

Document.Parent

Property **Parent** As **Documents**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Parent')} **Related Topics**

Document.SizeHeight

Property **SizeHeight** As Long

Description

The **SizeHeight** property returns or sets the height of a document in Corel PHOTO-PAINT, measured in pixels. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_SizeHeight')} **Related Topics**

Document.SizeWidth

Property **SizeWidth** As Long

Description

The **SizeWidth** property returns or sets the width of a document in Corel PHOTO-PAINT, measured in pixels. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_SizeWidth')} **Related Topics**

Document.Tools

Property **Tools** As **Tools**

Description

The **Tools** property returns a value associated with the **Tools** collection in a document in Corel PHOTO-PAINT. The Corel PHOTO-PAINT Toolbox contains tools for creating and manipulating objects and selections in your image. The zoom tools let you view specific areas of your image, and the shaping and painting tools let you modify your image. The Toolbox also contains tools that let you apply modifications interactively.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Tools')} **Related Topics**

Document.VirtualHeight

Property **VirtualHeight** As Long

Description

The **VirtualHeight** property returns or sets the virtual, or relative, height of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_VirtualHeight')} **Related Topics**

Document.VirtualWidth

Property **VirtualWidth** As Long

Description

The **VirtualWIdth** property returns or sets the virtual, or relative, width of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_VirtualWidth')} **Related Topics**

Document.Windows

Property **Windows** As **Windows**

Description

The **Windows** property returns a value associated with the **Windows** collection in a document in Corel PHOTO-PAINT.A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

This property returns a Read-Only value.

{button ,AL(`CLS_Document;FNC_Windows')} **Related Topics**

Document.Activate

Sub **Activate**()

Description

The **Activate** method opens a document in the main application window in Corel PHOTO-PAINT, if the window is not currently open, and makes the document active. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Activate')} **Related Topics**

Document.AddBackground

Sub **AddBackground**([ByVal Color As **Color** = 0])

Description

The **AddBackground** method adds a background, if there is not one, to a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Color	Sets the color of the added background. This value is optional and the default value is 0.

{button ,AL(`CLS_Document;FNC_AddBackground')} **Related Topics**

Document.ApplyICCProfile

Sub **ApplyICCProfile**(ByVal **FileName** As String)

Description

The **ApplyICCProfile** method applies the specified ICC profile to a document in Corel PHOTO-PAINT. The ICC profile is information based on the International Color Consortium (ICC), an organization that sets standards for device characterization.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
FileName	Specifies the computer location of the ICC profile.

{button ,AL(`CLS_Document;FNC_ApplyICCProfile')} **Related Topics**

Document.Checkpoint

Sub **Checkpoint**()

Description

The **Checkpoint** method stores the current state of a document in Corel PHOTO-PAINT. A checkpoint is a marked stage in your image's development to which you can return later.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Checkpoint')} **Related Topics**

Document.Close

Sub **Close**()

Description

The **Close** method closes the active document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Close')} **Related Topics**

Document.ConvertTo

Sub **ConvertTo**(ByVal **Mode** As **cdrImageMode**, [ByVal Flatten As Boolean = False])

Description

The **ConvertTo** method converts a document into another mode in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Mode	Sets the new mode in the document. This value returns cdrImageMode.
Flatten	Sets a True of False value that indicates whether or not to flatten the document. This value is optional and the default value is False. Values range from 0 to 99.

{button ,AL(`CLS_Document;FNC_ConvertTo')} **Related Topics**

Document.ConvertToBW

Sub **ConvertToBW**(ByVal **RenderType** As **cdrRenderType**, [ByVal Intensity As Long = 100], [ByVal Threshold As Long = 128], [ByVal Halftone As **cdrHalftoneType** = cdrHalftoneSquare (0)], [ByVal HalftoneAngle As Double = 45], [ByVal HalftoneSize As Long = 0])

Converts the document into the black-and-white mode

Member of **Document**

Parameters	Description
RenderType	Description of RenderType goes here (**in**)
Intensity	Description of Intensity goes here (**in**) Optional Default value = **100**
Threshold	Description of Threshold goes here (**in**) Optional Default value = **128**
Halftone	Description of Halftone goes here (**in**) Optional Default value = **cdrHalftoneSquare** (0)
HalftoneAngle	Description of HalftoneAngle goes here (**in**) Optional Default value = **45**
HalftoneSize	Description of HalftoneSize goes here (**in**) Optional Default value = **0**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Document;FNC_ConvertToBW')} **Related Topics**

Document.ConvertToDuotone

Sub **ConvertToDuotone**([ByVal DuotoneType As **cdrDuotoneType** = cdrMonotone (0)], [ByVal Color1 As **Color** = 0], [ByVal Color2 As **Color** = 0], [ByVal Color3 As **Color** = 0], [ByVal Color4 As **Color** = 0])

Converts the document into the duotone mode

Member of **Document**

Parameters	Description
DuotoneType	Description of DuotoneType goes here (**in**) Optional Default value = **cdrMonotone** (0)
Color1	Description of Color1 goes here (**in**) Optional Default value = **0**
Color2	Description of Color2 goes here (**in**) Optional Default value = **0**
Color3	Description of Color3 goes here (**in**) Optional Default value = **0**
Color4	Description of Color4 goes here (**in**) Optional Default value = **0**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Document;FNC_ConvertToDuotone')} **Related Topics**

Document.ConvertToMovie

Function **ConvertToMovie**() As Boolean

Description

The **ConvertToMovie** method sets a True or False value that indicates whether or not to convert a document into a movie in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_ConvertToMovie')} **Related Topics**

Document.ConvertToNTSC

Sub **ConvertToNTSC**()

Description

The **ConvertToNTSC** method converts the color in a 24-bit RGB image to colors that are suitable for NTSC standard television reproduction in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_ConvertToNTSC')} **Related Topics**

Document.ConvertToPAL

Sub **ConvertToPAL**()

Description

The **ConvertToPAL** method converts the color in a 24-bit RGB image to colors that are suitable for PAL standard television reproduction in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_ConvertToPAL')} **Related Topics**

Document.ConvertToPaletted

Sub **ConvertToPaletted**(ByVal **PaletteType** As **cdrImagePaletteType**, [ByVal Dithering As **cdrDitherType** = cdrDitherNone (0)], [ByVal DitherIntensity As Long = 100], [ByVal Smoothing As Long = 0], [ByVal NumColors As Long = 256], [ByVal Flatten As Boolean = False], [ByVal KeyColor As **Color** = 0], [ByVal KeyColorImportance As Long = 20], [ByVal SensitivityL As Long = 500], [ByVal SensitivityA As Long = 300], [ByVal SensitivityB As Long = 300], [ByVal PaletteFileName As String])

Converts the document into the paletted mode

Member of **Document**

Parameters	Description
PaletteType	Description of PaletteType goes here (**in**)
Dithering	Description of Dithering goes here (**in**) Optional Default value = **cdrDitherNone** (0)
DitherIntensity	Description of DitherIntensity goes here (**in**) Optional Default value = **100**
Smoothing	Description of Smoothing goes here (**in**) Optional Default value = **0**
NumColors	Description of NumColors goes here (**in**) Optional Default value = **256**
Flatten	Description of Flatten goes here (**in**) Optional Default value = **False**
KeyColor	Description of KeyColor goes here (**in**) Optional Default value = **0**
KeyColorImportance	Description of KeyColorImportance goes here (**in**) Optional Default value = **20**
SensitivityL	Description of SensitivityL goes here (**in**) Optional Default value = **500**
SensitivityA	Description of SensitivityA goes here (**in**) Optional Default value = **300**
SensitivityB	Description of SensitivityB goes here (**in**) Optional Default value = **300**
PaletteFileName	Description of PaletteFileName goes here (**in**) Optional

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Document;FNC_ConvertToPaletted')} **Related Topics**

Document.CreateGuide

Function **CreateGuide**(ByVal **Type** As **pntGuideType**, ByVal **PositionX** As Long, ByVal **PositionY** As Long) As **Guide**

Description

The **CreateGuide** method creates a new guideline in a document in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Type	Sets the guideline type of the new guideline. This value returns pntGuideType.
PositionX	Sets the horizontal position of the guideline, in document units.
PositionY	Sets the vertical position of the guideline, in document units.

{button ,AL(`CLS_Document;FNC_CreateGuide')} **Related Topics**

Document.CreatePalette

Function **CreatePalette**(ByVal **Name** As String, [ByVal FileName As String], [ByVal Overwrite As Boolean = False]) As **Palette**

Description

The **CreatePalette** method creates a new palette in a document in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Name	Sets a string value that names and uniquely identifies the new palette.
FileName	Sets the file name of the new palette. This value is optional.
Overwrite	Sets a True or False value that indicates if the new palette replaces another palette with the same file name. This value is optional and the default value is False.

{button ,AL(`CLS_Document;FNC_CreatePalette')} **Related Topics**

Document.CreatePaletteFromVisible

Function **CreatePaletteFromVisible**(ByVal **Name** As String, [ByVal FileName As String], [ByVal Overwrite As Boolean = False]) As **Palette**

Description

The **CreatePaletteFromVisible** method creates a new color palette from the color available in the visible layer in a document in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Name	Sets a string value that names and uniquely identifies the new palette.
FileName	Sets the file name of the new palette. This value is optional.
Overwrite	Sets a True or False value that indicates if the new palette replaces another palette with the same file name. This value is optional and the default value is False.

{button ,AL(`CLS_Document;FNC_CreatePaletteFromVisible')} **Related Topics**

Document.Crop

Sub **Crop**(ByVal **Left** As Long, ByVal **Top** As Long, ByVal **Width** As Long, ByVal **Height** As Long)

Description

The **Crop** method crops a document to a specified rectangle shape in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Left	Specifies the left position of the rectangle shape used to crop the document.
Top	Specifies the top position of the rectangle shape used to crop the document.
Width	Specifies the width of the rectangle shape used to crop the document.
Height	Specifies the height of the rectangle shape used to crop the document.

{button ,AL(`CLS_Document;FNC_Crop')} **Related Topics**

Document.CropToColor

Sub **CropToColor**(ByVal **Color** As **Color**, [ByVal Tolerance As **pntToleranceMode** = pntToleranceNormal (0)], [ByVal NormalLevel As Long = 0], [ByVal Hue As Long = 0], [ByVal Saturation As Long = 0], [ByVal Brightness As Long = 0])

Description

The **CropToColor** method crops a specific color border surrounding an image to the point where a different colored pixel is encountered in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Color	Specifies the color of the crop in the document.
Tolerance	Sets the mode of tolerance in the crop. This value returns pntToleranceMode. This value is optional and the default value is **pntToleranceNormal** (0).
Normal	Sets the normal level of the new color. This value is optional and the default value is 0.
Hue	Sets the hue level of the new color. This value is optional and the default value is 0. Hue is the property of a color that allows us to classify it by its name. For example, blue, green, and red are all hues.
Saturation	Sets the saturation level of the new color. This value is optional and the default value is 0. Saturation is the purity or vividness of a color, expressed as the absence of white. A color that has 100% saturation contains no white. A color with 0% saturation is a shade of gray.
Brightness	Sets the brightness level of the new color. This value is optional and the default value is 0. Brightness is the amount of light that is transmitted or reflected from a given pixel. In the HSB color model, brightness is a measure of how much white a color contains. In this case, a brightness value of 0 produces black and a brightness value of 255 produces white.

{button ,AL(`CLS_Document;FNC_CropToColor')} **Related Topics**

pntToleranceNormal0

pntToleranceHSB 1

Document.CropToMask

Sub **CropToMask**()

Description

The **CropToMask** method crops the selected image to the bounding box of the current mask in a document in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_CropToMask')} **Related Topics**

Document.DeskewCrop

Sub **DeskewCrop**(ByVal **Angle** As Double, ByVal **Width** As Long, ByVal **Height** As Long, ByVal **CenterX** As Long, ByVal **CenterY** As Long)

Description

The **DeskewCrop** method crops an image. If the angle is set to a value other than 0, the image is also deskewed in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Angle	Sets the angle of the crop.
Width	Sets the width of the crop.
Height	Sets the height of the crop.
CenterX	Sets the horizontal position of the center of the crop image.
CenterY	Sets the vertical position of the center of the crop image.

{button ,AL(`CLS_Document;FNC_DeskewCrop')} **Related Topics**

Document.Duplicate

Function **Duplicate**([ByVal Flatten As Boolean = False], ByVal **FileName** As String) As **Document**

Description

The **Duplicate** method creates a copy of the image and assigns it the specified name in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Flatten	Sets a True or False value that indicates whether or not to flatten the new document. This value is optional and the default value is False.
FileName	Specifies a new file name for the duplicated document.

{button ,AL(`CLS_Document;FNC_Duplicate')} **Related Topics**

Document.Flip

Sub **Flip**(ByVal **Axes** As **cdrFlipAxes**)

Description

The **Flip** method flips a document horizontally and/or vertically in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Axes	Sets the type of flip in the document. This value returns cdrFlipAxes.

{button ,AL(`CLS_Document;FNC_Flip')} **Related Topics**

Document.PaperSize

Sub **PaperSize**(ByVal **Width** As Long, ByVal **Height** As Long, [ByVal OffsetX As Long = 0], [ByVal OffsetY As Long = 0], [ByVal Color As **Color** = 0])

Description

The **PaperSize** method adjusts 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 in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Width	Sets the width of the paper.
Height	Sets the height of the paper.
OffsetX	Sets the horizontal offset value of the paper. This value is optional and the default value is False.
OffsetY	Sets the vertical offset value of the paper. This value is optional and the default value is False.
Color	Sets the paper color. This value is optional and the default value is 0.

{button ,AL(`CLS_Document;FNC_PaperSize')} **Related Topics**

Document.PrintOut

Sub **PrintOut**()

Description

The **PrintOut** method prints a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_PrintOut')} **Related Topics**

Document.Redo

Sub **Redo**()

Description

The **Redo** method reverses the last Undo operation performed in the active document of Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Redo')} **Related Topics**

Document.Repeat

Sub **Repeat**()

Description

The **Repeat** method repeats the last operations performed in the active document of Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Repeat')} **Related Topics**

Document.Resample

Sub **Resample**([ByVal Width As Long = 0], [ByVal Height As Long = 0], [ByVal AntiAlias As Boolean = True])

Description

The **Resample** method adjusts the dimensions and resolution of an image document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Width	Sets the new width of the document. This value is optional and the default value is 0. This value is measured in pixels.
Height	Sets the new height of the document. This value is optional and the default value is 0. This value is measured in pixels.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not document images with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Document;FNC_Resample')} **Related Topics**

Document.RestoreCheckpoint

Sub **RestoreCheckpoint**()

Description

The **RestoreCheckpoint** method restores the document as it was during the last checkpoint in Corel PHOTO-PAINT. A checkpoint is a marked stage in your image's development to which you can return later.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_RestoreCheckpoint')} **Related Topics**

Document.Revert

Sub **Revert**()

Description

The **Revert** method opens the last saved version of the document from disk in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Revert')} **Related Topics**

Document.Rotate

Sub **Rotate**(ByVal **Angle** As Double, [ByVal Clip As Boolean = False], [ByVal AntiAlias As Boolean = True], [ByVal Color As **Color** = 0])

Description

The **Rotate** method rotates the image document to a specified degree in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Angle	Sets the angle of rotation, in degrees, to rotate the document.
Clip	Sets a True or False value that indicates whether or not the document is clipped upon rotation. This value is optional and the default value is False.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not document images with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.
Color	Sets the color in the document. This value is optional and the default value is 0.

{button ,AL(`CLS_Document;FNC_Rotate')} **Related Topics**

Document.Save

Sub **Save**()

Description

The **Save** method saves the document using its current name and location in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Save')} **Related Topics**

Document.SaveAs

Sub **SaveAs**(ByVal **FileName** As String, ByVal **Filter** As **cdrFilter**, [ByVal Flatten As Boolean = False])

Description

The **SaveAs** method saves the document under a new name or in a different location in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
FileName	Sets the new file name for the saved document.
Filter	Sets the filter to use when saving the document. This value returns cdrFilter.
Flatten	Sets a True or False value that indicates whether or not to flatten the new document. This value is optional and the default value is False.

{button ,AL(`CLS_Document;FNC_SaveAs')} **Related Topics**

Document.SelectMovieFrames

Sub **SelectMovieFrames**(ByVal **Start** As Long, ByVal **End** As Long)

Description

The **SelectMovieFrames** method selects a range of movie frames in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Start	Specifies the first frame in the range of selection frames.
End	Specifies the last frame in the range of selection frames.

{button ,AL(`CLS_Document;FNC_SelectMovieFrames')} **Related Topics**

Document.SetResolution

Sub **SetResolution**(ByVal **NewDPI** As Long)

Description

The **SetResolution** methods sets the resolution of the active image document in dots per inch (dpi) in Corel PHOTO-PAINT. Resolution refers to the amount of detail and information that an image file contains, as well as the level of detail that an input, output or display device is capable of producing. When you work with bitmaps, resolution affects the quality of your final output and the file size.

A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
NewDPI	Sets a new resolution value in the document.

{button ,AL(`CLS_Document;FNC_SetResolution')} **Related Topics**

pntSplitRGB	0
pntSplitCMYK	1
pntSplitHSB	2
pntSplitHLS	3
pntSplitYIQ	4
pntSplitLAB	5

Document.Split

Function **Split**(ByVal **Mode** As **pntSplitMode**) As **Documents**

Description

The **Split** method separates an image document into the color channels corresponding to the specified color model in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

Parameters	Description
Mode	Sets the split mode in the document. This value returns pntSplitMode.

{button ,AL(`CLS_Document;FNC_Split')} **Related Topics**

Document.Undo

Sub **Undo**()

Description

The **Undo** method reverses the last operation performed in the active document of Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

{button ,AL(`CLS_Document;FNC_Undo')} **Related Topics**

Documents properties

Documents Legend

- Application
- Count
-
- Item
- Parent

Documents

Class **Documents**

Properties Referenced By

The **Documents** class defines the characteristics of **Documents** collection objects and describes the look and behavior of the collection objects through its properties and methods. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A document page is often referred to as a page. You can access any page in a multi-page document by clicking on a page tab in the Navigator of the Image Window. Each page in a multi-page document is accessible through the Navigator, located in the bottom-left corner of the Image Window. The Navigator allows you to move through your document quickly and shows the total number of pages in your image and the number of the page currently displayed. You can move to a page in your document by clicking that page number in the Navigator.

New documents use the preset options built into the Corel PHOTO-PAINT application. Every new document allows you to save settings so that new images opened in Corel PHOTO-PAINT open to the settings established in this Document page. These options include:

- Options set in the Options dialog box
- Toolbar settings
- Docker window
- Color palette

You can add, delete, and rename pages in documents as well as create web documents. Detailed information about each Corel PHOTO-PAINT document can be viewed and saved in a text file within a word processing application for future use.

{button ,AL(`CLS_Documents')} **Related Topics**

Documents.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub DocApp()
With Documents
    MsgBox .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Documents;FNC_Application')} **Related Topics**

Documents.Count

Property **Count** As Long

Description

The **Count** property returns the number of document objects in the **Documents** collection of Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

The **Count** property returns a Read-Only value.

Example

The following code example displays the number of documents in the **Documents** collection in a message box:

```
Sub DocsCount()
With Documents
    MsgBox .Count
End With
End Sub
```

{button ,AL(`CLS_Documents;FNC_Count')} **Related Topics**

Documents.Item

Property **Item**(ByVal **Index** As Long) As **Document**

Description

The **Item** property returns a value associated with the index number of a document in the **Documents** collection of Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

`Documents(2)` refers to the second effect in the **Documents** collection of Corel PHOTO-PAINT.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Documents.Item(2)` is the same as `Documents(2)` - they both reference the second document in the **Documents** collection.

You must reference a document in the collection by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each document in a **Documents** collection; it uniquely identifies each member of the collection.

Example

The following code example sets a new name for the first document in the **Documents** collection in Corel PHOTO-PAINT. The new document name displays in a message box:

```
Sub DocsItem()
With Documents
    .Item(1).Name = "Corel_PHOTO-PAINT_Document"
End With
End Sub
```

{button ,AL(`CLS_Documents;FNC_Item')} **Related Topics**

Documents.Parent

Property **Parent** As **Application**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the name of the **Documents** collection's parent layer in a message box:

```
Sub DocParent()
With Documents
    MsgBox .Parent.ActiveLayer.Name
End With
End Sub
```

{button ,AL(`CLS_Documents;FNC_Parent')} **Related Topics**

DropShadow properties

DropShadow Legend

- Angle
- Application
- Color
- Fade
- Feather
- FeatherEdge
- FeatherType
- Offset
- Opacity
- Parent
- PerspectiveStretch
- Type

DropShadow methods

DropShadow Legend

- Combine
- Delete
- Split

cdrFeatherInside=0

cdrFeatherMiddle=1

cdrFeatherOutside=2

cdrFeatherAverage=3

cdrEdgeLinear=0

cdrEdgeSquared=1

cdrEdgeFlat=2

cdrEdgeInverseSquared=3

cdrEdgeMesa=4

cdrEdgeGaussian=5

cdrDropShadowFlat=0

cdrDropShadowBottom=1

cdrDropShadowTop=2

cdrDropShadowLeft=3

cdrDropShadowRight=4

DropShadow

Class **DropShadow**

Properties Methods Referenced By

The **DropShadow** class defines the characteristics of drop shadow objects and describes the look and behavior of the objects through its properties and methods.

By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

Additionally, you can edit drop shadows to customize their effects. For example, you can change the color of a drop shadow. The feathering and opacity values you specify are added cumulatively to the transparency attributes already contained in the original object. When you choose a feather direction other than Average, you can also change the shape of the feathered pixels by choosing Linear or Curved from the Feather Type list box. When you create a drop shadow of an object, an icon appears next to the object thumbnail in the Objects Docker window. The drop shadow of an object automatically changes to mirror changes made to the shape or transparency of the object.

You can also change the opacity of a shadow in the Image Window by dragging the opacity node on the direction arrow. You can also choose a color for the shadow by double-clicking the outlined direction node of the shadow.

{button ,AL(`CLS_DropShadow')} **Related Topics**

DropShadow.Angle

Property **Angle** As Double

Description

The **Angle** returns or sets a value that specifies the angle at which the shadow lies in relation to the object. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

{button ,AL(`CLS_DropShadow;FNC_Angle')} **Related Topics**

DropShadow.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Application')} **Related Topics**

DropShadow.Color

Property **Color** As **Color**

Description

The **Color** property returns the color of a drop shadow in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Color')} **Related Topics**

DropShadow.Fade

Property **Fade** As Long

Description

The **Fade** property returns the fade value of a drop shadow in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

The Fade property specifies the percentage by which the shadow fades as it moves away from the object. You cannot change the fade level of a drop shadow that has a Flat perspective.

This property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Fade')} **Related Topics**

DropShadow.Feather

Property **Feather** As Long

Description

The **Feather** property returns the feather length for a drop shadow in Corel PHOTO-PAINT. You can change a drop shadow's feathering edge style, the feathering direction, as well as the level of feathering. You can type values between 0 and 100. Low values create a more subtle feathering effect, while high values create a more pronounced effect.

Feathering is the gradual blending of pixels between a selection or an object and the surrounding background. Feathering produces a softer, more natural-looking edge.

By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Feather')} **Related Topics**

DropShadow.FeatherEdge

Property **FeatherEdge** As **pntEdgeType**

Description

The **FeatherEdge** property returns the feather edge for a drop shadow in Corel PHOTO-PAINT. You can change a drop shadow's feathering edge style, the feathering direction, as well as the level of feathering. You can type values between 0 and 100. Low values create a more subtle feathering effect, while high values create a more pronounced effect.

Feathering is the gradual blending of pixels between a selection or an object and the surrounding background. Feathering produces a softer, more natural-looking edge.

By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns a Read-Only value of cdrEdgeType.

{button ,AL(`CLS_DropShadow;FNC_FeatherEdge')} **Related Topics**

DropShadow.FeatherType

Property **FeatherType** As **pntFeatherType**

Description

The **FeatherType** property returns the feather type for a drop shadow in Corel PHOTO-PAINT. You can change a drop shadow's feathering edge style, the feathering direction, as well as the level of feathering. You can type values between 0 and 100. Low values create a more subtle feathering effect, while high values create a more pronounced effect.

Feathering is the gradual blending of pixels between a selection or an object and the surrounding background. Feathering produces a softer, more natural-looking edge.

By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns Read-Only value of cdrFeatherType.

{button ,AL(`CLS_DropShadow;FNC_FeatherType')} **Related Topics**

DropShadow.Offset

Property **Offset** As Double

Description

The **Offset** property returns the flat offset value of a drop shadow in Corel PHOTO-PAINT. The offset value sets the distance between the drop shadow effect and its control object, based on the units of measurement of the image.

By adding drop shadows, you create the illusion of depth between objects. You can add drop shadows to most objects (or groups of objects) you create using Corel PHOTO-PAINT, including Artistic text, Paragraph text, and bitmap images. However, you cannot add drop shadows to link groups such as blended objects, contoured objects, beveled objects, extruded objects, or other drop shadows.

This property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Offset')} **Related Topics**

DropShadow.Opacity

Property **Opacity** As Long

Description

The **Opacity** property returns the opacity value of a drop shadow in Corel PHOTO-PAINT. Opacity determines the intensity of a drop shadow. You can type values between 0 and 100. Low values create a less opaque drop shadow, while high values create a more opaque drop shadow.

By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Opacity')} **Related Topics**

DropShadow.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_Parent')} **Related Topics**

DropShadow.PerspectiveStretch

Property **PerspectiveStretch** As Double

Description

The **PerspectiveStretch** returns the feather perspective stretch value in a drop shadow in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns a Read-Only value.

{button ,AL(`CLS_DropShadow;FNC_PerspectiveStretch')} **Related Topics**

DropShadow.Type

Property **Type** As **pntDropShadowType**

Description

The **Type** property returns the drop shadow type in a drop shadow in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

This property returns cdrDropShadowType.

{button ,AL(`CLS_DropShadow;FNC_Type')} **Related Topics**

DropShadow.Combine

Sub **Combine**()

Description

The **Combine** method combines a drop shadow effect with an object in Corel PHOTO-PAINT. When you combine objects, you permanently create one object from multiple objects. You can also combine objects with the background to decrease the file size of an image. When you combine objects with the background, they no longer float above the rest of the image, and they cannot be selected or changed individually.

By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

{button ,AL(`CLS_DropShadow;FNC_Combine')} **Related Topics**

DropShadow.Delete

Sub **Delete**()

Description

The **Delete** method deletes a drop shadow in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

{button ,AL(`CLS_DropShadow;FNC_Delete')} **Related Topics**

DropShadow.Split

Function **Split**() As **Layer**

Description

The **Split** method creates a new object with a drop shadow in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

{button ,AL(`CLS_DropShadow;FNC_Split')} **Related Topics**

Floater properties

Floater Legend

- Application

- Parent
  - PositionX
  - PositionY

- SizeHeight

- SizeWidth

Floater methods

Floater Legend

- CopyToClipboard
- CutToClipboard
- Defloat
- Delete
- GetPosition
- Move
- SetPosition

Floater

Class **Floater**

Properties Methods Referenced By

The **Floater** class defines the characteristics of floater objects and describes the look and behavior of the application through its properties and methods.

A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

{button ,AL(`CLS_Floater')} **Related Topics**

Floater.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Floater;FNC_Application')} **Related Topics**

Floater.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Floater;FNC_Parent')} **Related Topics**

Floater.PositionX

Property **PositionX** As Long

Description

The **PositionX** property returns or sets the horizontal position of a floating object's center in Corel PHOTO-PAINT. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

{button ,AL(`CLS_Floater;FNC_PositionX')} **Related Topics**

Floater.PositionY

Property **PositionY** As Long

Description

The **PositionY** property returns or sets the vertical position of a floating object's center in Corel PHOTO-PAINT. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

{button ,AL(`CLS_Floater;FNC_PositionY')} **Related Topics**

Floater.SizeHeight

Property **SizeHeight** As Long

Description

The **SizeHeight** property returns or sets the height of a floater object in Corel PHOTO-PAINT, measured in documents units. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

This property returns a Read-Only value.

{button ,AL(`CLS_Floater;FNC_SizeHeight')} **Related Topics**

Floater.SizeWidth

Property **SizeWidth** As Long

Description

The **SizeWidth** property returns or sets the width of a floater object in Corel PHOTO-PAINT, measured in documents units. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

This property returns a Read-Only value.

{button ,AL(`CLS_Floater;FNC_SizeWidth')} **Related Topics**

Floater.CopyToClipboard

Sub **CopyToClipboard**()

Description

The **CopyToClipboard** method copies the contents of a floating selection in Corel PHOTO-PAINT into the system clipboard. The clipboard is a temporary storage area that is used to hold cut or copied information. The clipboard is always present in Windows and it stores information until it is replaced by another object or selection that has been cut or copied, or until you quit the current session of Windows.

A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

Copying to the clipboard leaves the floating selection unchanged in Corel PHOTO-PAINT.

{button ,AL(`CLS_Floater;FNC_CopyToClipboard')} **Related Topics**

Floater.CutToClipboard

Sub **CutToClipboard**()

Description

The **CutToClipboard** method cuts the contents of a floating selection in Corel PHOTO-PAINT into the system clipboard. The clipboard is a temporary storage area that is used to hold cut or copied information. The clipboard is always present in Windows and it stores information until it is replaced by another object or selection that has been cut or copied, or until you quit the current session of Windows.

A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

Cutting to the clipboard removes the floating selection from Corel PHOTO-PAINT.

{button ,AL(`CLS_Floater;FNC_CutToClipboard')} **Related Topics**

Floater.Defloat

Sub **Defloat**()

Description

The **Defloat** method defloats the floater object in Corel PHOTO-PAINT. Defloating merges the pixels of the floating selection and its underlying image.

A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

{button ,AL(`CLS_Floater;FNC_Defloat')} **Related Topics**

Floater.Delete

Sub **Delete**()

Description

The **Delete** method deletes the floater object in Corel PHOTO-PAINT. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

{button ,AL(`CLS_Floater;FNC_Delete')} **Related Topics**

Floater.GetPosition

Sub **GetPosition**(ByRef **PositionX** As Long, ByRef **PositionY** As Long)

Description

The **GetPosition** method gets the center point of a floater object in Corel PHOTO-PAINT. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

Parameters	Description
PositionX	Gets the horizontal position of the floater's center point, in document units.
PositionY	Gets the vertical position of the floater's center point, in document units.

{button ,AL(`CLS_Floater;FNC_GetPosition')} **Related Topics**

Floater.Move

Sub **Move**(ByVal **Dx** As Long, ByVal **Dy** As Long)

Description

The **Move** method changes the position of a floater object by moving it a specified distance from its original location in Corel PHOTO-PAINT. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

Parameters	Description
Dx	The **Dx** parameter specifies the horizontal distance to move a floater with the **Move** method. Delta refers to a limited incremental value in a variable.
Dy	The **Dy** parameter specifies the vertical distance to move a floater with the **Move** method. Delta refers to a limited incremental value in a variable.

{button ,AL(`CLS_Floater;FNC_Move')} **Related Topics**

Floater.SetPosition

Sub **SetPosition**(ByVal **PositionX** As Long, ByVal **PositionY** As Long)

Description

The **SetPosition** method sets the center point of a floater object in Corel PHOTO-PAINT. A floater is a floating selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

Parameters	Description
PositionX	Sets the horizontal position of the floater's center point, in document units.
PositionY	Sets the vertical position of the floater's center point, in document units.

{button ,AL(`CLS_Floater;FNC_SetPosition')} **Related Topics**

Frame properties

Frame Legend

- Application

- Background
  Delay

- Index

- Parent

Frame methods

Frame Legend

- Activate
- Delete
- Move

Frame

Class **Frame**

Properties Methods Referenced By

The **Frame** class defines the characteristics of frame objects and describes the look and behavior of the application through its properties and methods. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

The Frame filter supports all color modes except 48-bit RGB, 16-bit grayscale, paletted, and black-and-white.

{button ,AL(`CLS_Frame')} **Related Topics**

Frame.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Frame;FNC_Application')} **Related Topics**

Frame.Background

Property **Background** As **Background**

Description

The **Background** property returns a value associated with the background of a frame in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

This property returns a Read-Only value.

{button ,AL(`CLS_Frame;FNC_Background')} **Related Topics**

Frame.Delay

Property **Delay** As Long

Description

The **Delay** property returns or sets the frame rate of a frame in Corel PHOTO-PAINT. Frame rate is the amount of time that a movie frame appears on the screen. You can assign a display length to individual frames or to all the frames in a movie at once. Changing the frame rate lets you increase or decrease the speed of moving objects from one frame to another.

A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

{button ,AL(`CLS_Frame;FNC_Delay')} **Related Topics**

Frame.Index

Property **Index** As Long

Description

The **Index** property returns a value associated with a frame object in the **Frames** collection of Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

This property returns a Read-Only value.

{button ,AL(`CLS_Frame;FNC_Index')} **Related Topics**

Frame.Parent

Property **Parent** As **Frames**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Frame;FNC_Parent')} **Related Topics**

Frame.Activate

Sub **Activate**()

Description

The **Activate** method activates a frame in Corel PHOTO-PAINT, making it the active frame in the application. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

{button ,AL(`CLS_Frame;FNC_Activate')} **Related Topics**

Frame.Delete

Sub **Delete**()

Description

The **Delete** method deletes a frame in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

{button ,AL(`CLS_Frame;FNC_Delete')} **Related Topics**

Frame.Move

Sub **Move**(ByVal **Index** As Long)

Description

The **Move** method moves the frame to a new position in the **Frames** collection by specifying a new index value for the frame in Corel PHOTO-PAINT.A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
Index	Specifies the new index number of the moved frame in the **Frames** collection.

{button ,AL(`CLS_Frame;FNC_Move')} **Related Topics**

Frames properties

Frames Legend

- Application
- Count
-
- Item
- Parent

Frames methods

Frames Legend

- Activate
- Delete
- Insert
- InsertFromFile
- Move
- SetDelay

Frames

Class **Frames**

Properties Methods Referenced By

The **Frames** class defines the characteristics of **Frames** collection objects and describes the look and behavior of the application through its properties and methods. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

The Frame filter supports all color modes except 48-bit RGB, 16-bit grayscale, paletted, and black-and-white.

{button ,AL(`CLS_Frames')} **Related Topics**

Frames.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Frames;FNC_Application')} **Related Topics**

Frames.Count

Property **Count** As Long

Description

The **Count** property returns the number of frames in the **Frames** collection of Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

The **Count** property returns a Read-Only value.

{button ,AL(`CLS_Frames;FNC_Count')} **Related Topics**

Frames.Item

Property **Item**(ByVal **Index** As Long) As **Frame**

Description

The **Item** property returns a value associated with the index number of a frame object in the **Frames** collection of Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

`Frames(2)` refers to the second frame object in the **Frames** collection of Corel PHOTO-PAINT.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Frames.Item(2)` is the same as `Frames(2)` - they both reference the second frame object in the **Frames** collection.

You must reference a frame in the collection by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each object in a **Frames** collection; it uniquely identifies each member of the collection.

{button ,AL(`CLS_Frames;FNC_Item')} **Related Topics**

Frames.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Frames;FNC_Parent')} **Related Topics**

Frames.Activate

Sub **Activate**(ByVal **FrameReference** As **pntFrameReference**)

Description

The **Activate** method activates a frame in the **Frames** collection of Corel PHOTO-PAINT, making it the active frame in the application. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
FrameReference	Sets the frame to be activated with the **Activate** method. This value returns pntFrameReference.

{button ,AL(`CLS_Frames;FNC_Activate')} **Related Topics**

pntFirstFrame	0
pntLastFrame	1
pntNextFrame	2
pntPreviousFrame	3

Frames.Delete

Sub **Delete**(ByVal **StartIndex** As Long, ByVal **EndIndex** As Long)

Description

The **Delete** method deletes a range of frames in Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
StartIndex	Specifies the index number of the first frame in a range of frames that are deleted with the **Delete** method.
EndIndex	Specifies the index number of the last frame in a range of frames that are deleted with the **Delete** method.

{button ,AL(`CLS_Frames;FNC_Delete')} **Related Topics**

Frames.Insert

Function **Insert**([ByVal Index As Long = 0], [ByVal Number As Long = 1], [ByVal Background As **pntFrameBackground** = pntFrameFill (0)], [ByVal Color As **Color** = 0], [ByVal Frame As **Frame** = 0]) As **Frame**

Description

The **Insert** method inserts a new frame in the **Frames** collection of Corel PHOTO-PAINT, at a specified index point in the collection. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
Index	Sets the index placeholder in the collection to insert the frame. This value is optional and the default value is 0.
Number	Sets the number of inserted frames. This value is optional and the default value is 1.
Background	Sets the background of the inserted frame. This value returns pntFrameBackground . This value is optional and the default value is pntFrameFill (0).
Color	Sets the color of the inserted frame. This value is optional and the default value is 0.
Frame	Sets the frame to be inserted with the **Insert** method. This value is optional and the default value is 0.

{button ,AL(`CLS_Frames;FNC_Insert')} **Related Topics**

pntFrameFill	0
pntFrameCopy	1
pntFrameCopyActive	2

Frames.InsertFromFile

Function **InsertFromFile**(ByVal **FileName** As String, [ByVal Index As Long = 0], [ByVal Filter As **cdrFilter** = cdrAutoSense (0)], [ByVal LoadMode As **pntLoadMode** = pntLoadAll (0)], [ByVal Left As Long = 0], [ByVal Top As Long = 0], [ByVal Width As Long = 0], [ByVal Height As Long = 0]) As **Frame**

Description

The **Insert** method inserts a frame from a file into the **Frames** collection of Corel PHOTO-PAINT. A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
FileName	Secifies the file name of the file containing the frame that is inserted with the **InsertFromFile** method.
Index	Sets the index placeholder in the collection to insert the frame. This value is optional and the default value is 0.
Filter	Specifies the filter to use when inserting the frame. This value returns cdrFilter. This value is optional and the default value is cdrAutoSense (0).
LoadMode	Sets the load mode of the inserted frame. This value returns pntLoadMode. This value is optional and the default value is pntLoadAll (0).
Left	Sets the left position of the inserted frame. This value is optional and the default value is 0.
Top	Sets the top position of the inserted frame. This value is optional and the default value is 0.
Width	Sets the width of the inserted frame. This value is optional and the default value is 0.
Height	Sets the height of the inserted frame. This value is optional and the default value is 0.

{button ,AL(`CLS_Frames;FNC_InsertFromFile')} **Related Topics**

pntLoadAll	0
pntLoadCrop	1
pntLoadResample	2

Frames.Move

Sub **Move**(ByVal **StartIndex** As Long, ByVal **EndIndex** As Long, ByVal **ToIndex** As Long)

Description

The **Move** method moves a range of frames within the Frames collection of Corel PHOTO-PAINT, by specifying a new index for the moved frame. Moves the specified frame to a given index.

A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
StartIndex	Specifies the index number of the first frame in a range of frames that are moved with the **Move** method.
EndIndex	Specifies the index number of the last frame in a range of frames that are moved with the **Move** method.
ToIndex	Specifies the new index number of the first frame in the range of frames that are moved with the **Move** method.

{button ,AL(`CLS_Frames;FNC_Move')} **Related Topics**

Frames.SetDelay

Sub **SetDelay**(ByVal **StartIndex** As Long, ByVal **EndIndex** As Long, ByVal **Delay** As Long)

Description

The **SetDelay** method sets the frame rate for frames in Corel PHOTO-PAINT. Frame rate is the amount of time that a movie frame appears on the screen. You can assign a display length to individual frames or to all the frames in a movie at once. Changing the frame rate lets you increase or decrease the speed of moving objects from one frame to another.

A frame object frames your image using one of 150 preset frames, another image, or an area defined by a mask. You can preview, select, and apply multiple frames on your image. You can also change the color, opacity, and alignment of a frame.

Creative special effects filters use a variety of shapes and textures to transform your image into abstract art. Use craft items, crystals, fabric, glass, game pieces, frames, whirlpools, or raindrops as the foundation for creating something wild or wonderful with your image.

You can insert empty frames into a movie, or you can insert frames that have been copied from other frames. You can insert up to 100 frames into a movie at a time. You can insert more than 100 frames into a movie, but not all at once.

Parameters	Description
StartIndex	Specifies the index number of the first frame in a range of frames that have their delay value set with the **SetDelay** method.
EndIndex	Specifies the index number of the last frame in a range of frames that have their delay value set with the **SetDelay** method.
Delay	Sets the delay time for the selected frames. The delay value will measure the amount of time passed when moving from one frame to another in a movie.

{button ,AL(`CLS_Frames;FNC_SetDelay')} **Related Topics**

GlobalDocument events

GlobalDocument

NewDocument
OpenDocument
Quit
Start

GlobalDocument

Class **GlobalDocument**

Events Referenced By

{button ,AL(`CLS_GlobalDocument')} **Related Topics**

GlobalDocument.NewDocument

Event **NewDocument**(ByVal **Document** As **Document**)

Member of **GlobalDocument**

Parameters	Description
Document	Description of Document goes here (**in**)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_GlobalDocument;FNC_NewDocument')} **Related Topics**

GlobalDocument.OpenDocument

Event **OpenDocument**(ByVal **Document** As **Document**)

Member of **GlobalDocument**

Parameters	Description
Document	Description of Document goes here (**in**)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_GlobalDocument;FNC_OpenDocument')} **Related Topics**

GlobalDocument.Quit

Event **Quit**()

Member of **GlobalDocument**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_GlobalDocument;FNC_Quit')} **Related Topics**

GlobalDocument.Start

Event **Start**()

Member of **GlobalDocument**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_GlobalDocument;FNC_Start')} **Related Topics**

Guide properties

Guide Legend

- Application

- Index

- Parent
  - PositionX
  - PositionY
  - Selected

- Type

Guide methods

Guide Legend

Delete

Guide

Class **Guide**

Properties Methods Referenced By

The **Guide** class defines the characteristics of guideline objects and describes the look and behavior of the objects through its properties and methods. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

You can change the color of the guidelines to make them stand out against the image background.

The Snap To Guidelines command makes guidelines magnetic. This means that when you move an object or selection close to a guideline, the object or selection automatically jumps to align with that line. You can set the sensitivity of this feature so that if you move an object or selection within the specified number of pixels of a guideline, the object or selection snaps to that line.

You can set up guidelines by specifying precise locations on the Horizontal and Vertical pages in the Options dialog box. The values that you specify in the Options dialog box are measured in the same units as the rulers, and represent the location of the guidelines relative to the rulers' settings. You can also delete and move guidelines.

{button ,AL(`CLS_Guide')} **Related Topics**

Guide.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main application object in Corel PHOTO-PAINT. Corel PHOTO-PAINT is a bitmap-based image editing and painting application that lets you retouch photographs, edit images and video files, and create original artwork. Corel PHOTO-PAINT provides special effects filters, painting, masking, and object handling tools.

Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

This property returns a Read-Only value.

{button ,AL(`CLS_Guide;FNC_Application')} **Related Topics**

Guide.Index

Property **Index** As Long

Description

The **Index** property returns a value associated with a guideline object in the **Guides** collection of Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

`Guides(5)` refers to the fifth guideline in the **Guides** collection of Corel PHOTO-PAINT.

The **Index** property returns a Read-Only value.

{button ,AL(`CLS_Guide;FNC_Index')} **Related Topics**

Guide.Parent

Property **Parent** As **Guides**

Description

The **Parent** property returns a value associated with the parent object of a guideline in Corel PHOTO-PAINT. A parent is an object that retains its shape but contains the color or texture of the child objects that are clipped to it in a clipping group. An object is always the parent to the objects that are listed above it in the Objects Docker window.

Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

This property returns a Read-Only value.

{button ,AL(`CLS_Guide;FNC_Parent')} **Related Topics**

Guide.PositionX

Property **PositionX** As Long

Description

The **PositionX** property returns or sets the horizontal position of a guide in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

{button ,AL(`CLS_Guide;FNC_PositionX')} **Related Topics**

Guide.PositionY

Property **PositionY** As Long

Description

The **PositionY** property returns or sets the vertical position of a guide in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

{button ,AL(`CLS_Guide;FNC_PositionY')} **Related Topics**

Guide.Selected

Property **Selected** As Boolean

Description

The **Selected** property returns or sets a True or False value that indicates whether or not a guideline is selected in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

{button ,AL(`CLS_Guide;FNC_Selected')} **Related Topics**

Guide.Type

Property **Type** As **pntGuideType**

Description

The **Type** property returns the specific kind of guideline used in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

The **Type** property returns a Read-Only value and returns pntGuideType.

{button ,AL(`CLS_Guide;FNC_Type')} **Related Topics**

pntGuideVertical	0
pntGuideHorizontal	1

Guide.Delete

Sub **Delete**()

Description

The **Delete** method deletes a guideline in Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

{button ,AL(`CLS_Guide;FNC_Delete')} **Related Topics**

Guides properties

Guides Legend

- Application
- Count
-
- Item
- Parent

Guides

Class **Guides**

Properties Referenced By

The **Guides** class defines the characteristics of guideline collection objects and describes the look and behavior of the objects through its properties and methods. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

You can change the color of the guidelines to make them stand out against the image background.

The Snap To Guidelines command makes guidelines magnetic. This means that when you move an object or selection close to a guideline, the object or selection automatically jumps to align with that line. You can set the sensitivity of this feature so that if you move an object or selection within the specified number of pixels of a guideline, the object or selection snaps to that line.

You can set up guidelines by specifying precise locations on the Horizontal and Vertical pages in the Options dialog box. The values that you specify in the Options dialog box are measured in the same units as the rulers, and represent the location of the guidelines relative to the rulers' settings. You can also delete and move guidelines.

{button ,AL(`CLS_Guides')} **Related Topics**

Guides.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main application object in Corel PHOTO-PAINT. Corel PHOTO-PAINT is a bitmap-based image editing and painting application that lets you retouch photographs, edit images and video files, and create original artwork. Corel PHOTO-PAINT provides special effects filters, painting, masking, and object handling tools.

Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

This property returns a Read-Only value.

{button ,AL(`CLS_Guides;FNC_Application')} **Related Topics**

Guides.Count

Property **Count** As Long

Description

The **Count** property counts the number of guidelines in the **Guides** collection of Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

This property returns a Read-Only value.

{button ,AL(`CLS_Guides;FNC_Count')} **Related Topics**

Guides.Item

Property **Item**(ByVal **Index** As Long) As **Guide**

Description

The **Item** property returns a value associated with the index number of a guideline object in the **Guides** collection of Corel PHOTO-PAINT. Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

`Guides(5)` refers to the fifth guideline in the collection. The **Item** property returns a Read-Only value. The **Item** property is the default property and may be omitted when referencing items in the collection.

You must reference an arrowhead in the collection by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each object in the **Guides** collection; it uniquely identifies each member of the collection.

{button ,AL(`CLS_Guides;FNC_Item')} **Related Topics**

Guides.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the parent object of a guideline collection object in Corel PHOTO-PAINT. A parent is an object that retains its shape but contains the color or texture of the child objects that are clipped to it in a clipping group. An object is always the parent to the objects that are listed above it in the Objects Docker window.

Guidelines are non-printing lines used to align objects. You can place guidelines anywhere in the Image Window by dragging them from the rulers or using the Guidelines Setup command in the Options dialog box.

Guidelines are vertical or horizontal lines that you can place anywhere in the Image Window to help you align and position image components. You can manually drag as many guidelines as you like from the ruler, or you can create them by using the Horizontal and Vertical pages in the Options dialog box. Once created, you can select, move, and delete guidelines in the Image Window. Snapping objects and selections to guidelines lets you automatically align objects and selections with the guidelines. When you save an image in Corel PHOTO-PAINT, the guidelines are saved too.

This property returns a Read-Only value.

{button ,AL(`CLS_Guides;FNC_Parent')} **Related Topics**

Layer properties

Layer Legend

- Active

- Application
  - ClipToParent

- Grouped

- LayerGroup
  - MergeMode
  - Name

- ObjectLayer
  - Opacity

- Parent
  - PositionX
  - PositionY
  - Selected
  - SizeHeight
  - SizeWidth

- Type
  - URLAddress
  - URLComment
  - URLRegion
  - Visible
  - ZOrder

Layer methods

Layer Legend

- Activate
- AffineDistort
- AlignToDocument
- AlignToDocumentCenter
- AlignToGrid
- AlignToLayer
- AlignToLayerRange
- AlignToPoint
- Copy
- CreateFloater
- CreateMask
- CreatePalette
- CreateSelection
- CropToMask
- Cut
- Delete
- Distort
- Duplicate
- Feather
- Flip
- GetPosition
- GetSize
- IsInFrontOf
- Merge
- Move
- OrderBackOf
- OrderBackOne
- OrderForwardOne
- OrderFrontOf
- OrderToBack
- OrderToFront
- PasteFloater
- Rotate
- SetPosition
- SetSize
- Skew
- Stretch
- Threshold
- Ungroup
- UngroupAll

Layer

Class **Layer**

Properties Methods Referenced By

The **Layer** class defines the characteristics of layer objects and describes the look and behavior of the objects through its properties and methods.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

The layering feature of Corel PHOTO-PAINT gives you added flexibility for organizing and editing the objects in your images. You can divide an image into multiple layers, each containing a portion of the image's contents. Together, layers act as a hierarchy that helps determine the vertical arrangement of an image's components. In this arrangement (the stacking order), objects on the top layer always overlay objects on the layer below.

{button ,AL(`CLS_Layer')} **Related Topics**

Layer.Active

Property **Active** As Boolean

Description

The **Active** property returns a True or False value indicating the status of a layer in Corel PHOTO-PAINT. If the layer is active, a value of true is returned, indicating that it is currently in use. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

If the window is inactive, the return value is false, indicating that the layer is currently open but not in use.

Use the **Activate** method to change the inactive status of the main application window.

The **Active** property returns a Read-Only value.

{button ,AL(`CLS_Layer;FNC_Active')} **Related Topics**

Layer.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub LayerApp()
With ActiveLayer
   MsgBox .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Layer;FNC_Application')} **Related Topics**

Layer.ClipToParent

Property **ClipToParent** As Boolean

Description

The **ClipToParent** property returns or sets a True or False value that indicates whether or not to clip the pixels of a child layer to the shape of its parent layer. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_ClipToParent')} **Related Topics**

Layer.Grouped

Property **Grouped** As Boolean

Description

The **Grouped** property returns a True or False value that indicates whether or not a layer is grouped in Corel PHOTO-PAINT. Grouping objects lets you move, delete, or transform objects as a single entity. When you group objects, a highlighting box appears around them in the Image Window and a thick, black line connects the objects in the Objects Docker window.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_Layer;FNC_Grouped')} **Related Topics**

Layer.LayerGroup

Property **LayerGroup** As **LayerRange**

Description

The **LayerGroup** property returns a value associated with a grouped layer range of Corel PHOTO-PAINT. Grouping objects lets you move, delete, or transform objects as a single entity. When you group objects, a highlighting box appears around them in the Image Window and a thick, black line connects the objects in the Objects Docker window.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_Layer;FNC_LayerGroup')} **Related Topics**

Layer.MergeMode

Property **MergeMode** As **pntMergeMode**

Description

The **MergeMode** property returns or sets the mode merge in a layer in Corel PHOTO-PAINT. A merge mode is an editing state that determines how the selected paint, object, or fill color combines with other colors in the image. For example, if you apply color or merge an object into the background using the Normal merge mode, the selected color replaces the original color in the image. If you apply color or merge an object into the background using the Add merge mode, the paint and paper colors are combined to produce a brighter color.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a value of pntMergeMode.

{button ,AL(`CLS_Layer;FNC_MergeMode')} **Related Topics**

Layer.Name

Property **Name** As String

Description

The **Name** property returns or sets a string value that names a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Example

The following code example displays the name of the active layer in a message box:

```
Sub Test()
    MsgBox "Active Layer is " & ActiveLayer.Name
End Sub
```

{button ,AL(`CLS_Layer;FNC_Name')} **Related Topics**

Layer.ObjectLayer

Property **ObjectLayer** As **ObjectLayer**

Description

The **ObjectLayer** property returns a value associated with the object layer in Corel PHOTO-PAINT. An object layer is the layer containing an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This value returns a Read-Only value.

{button ,AL(`CLS_Layer;FNC_ObjectLayer')} **Related Topics**

Layer.Opacity

Property **Opacity** As Long

Description

The **Opacity** property returns or set the level of opacity in a layer in Corel PHOTO-PAINT. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_Opacity')} **Related Topics**

Layer.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example counts the number of layer in the active layer's parent **Layers** collection in Corel PHOTO-PAINT:

```
Sub Layer()
MsgBox ActiveLayer.Parent.Count
End Sub
```

{button ,AL(`CLS_Layer;FNC_Parent')} **Related Topics**

Layer.PositionX

Property **PositionX** As Long

Description

The **PositionX** property returns or set the horizontal position of a layer's center in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_PositionX')} **Related Topics**

Layer.PositionY

Property **PositionY** As Long

Description

The **PositionY** property returns or set the vertical position of a layer's center in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_PositionY')} **Related Topics**

Layer.Selected

Property **Selected** As Boolean

Description

The **Selected** property returns or set s a True or False value that indicates whether or not a layer is selected in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_Selected')} **Related Topics**

Layer.SizeHeight

Property **SizeHeight** As Long

Description

The **SizeHeight** property returns or sets the height of a layer in Corel PHOTO-PAINT, measured in documents units. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_SizeHeight')} **Related Topics**

Layer.SizeWidth

Property **SizeWidth** As Long

Description

The **SizeWidth** property returns or sets the width of a layer in Corel PHOTO-PAINT, measured in documents units. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_SizeWidth')} **Related Topics**

Layer.Type

Property **Type** As **pntLayerType**

Description

The **Type** property returns the layer type of a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property is Read-Only and returns a value of pntLayerType.

{button ,AL(`CLS_Layer;FNC_Type')} **Related Topics**

pntObjectLayer	0
pntLensLayer	1

Layer.URLAddress

Property **URLAddress** As String

Description

The **URLAddress** property returns or sets the web address associated with a URL region in a layer in Corel PHOTO-PAINT. A URL is a unique address that defines where a Web page is located on the Internet. You can assign Internet addresses, or Uniform Resource Locators (URLs), to the objects in your images by defining clickable areas. When you click an area in your image that has an Internet address associated with it, you are linked to the specified URL. A clickable area can be a polygon that closely follows an object's shape, a rectangle that matches an object's highlighting box, an oval that fits within an object's highlighting box, or a circle that has a radius equal to the object's longest dimension from its center to its edges.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_URLAddress')} **Related Topics**

Layer.URLComment

Property **URLComment** As String

Description

The **URLComment** property returns or sets a string value that describes a URL region in a layer in Corel PHOTO-PAINT. A URL is a unique address that defines where a Web page is located on the Internet. You can assign Internet addresses, or Uniform Resource Locators (URLs), to the objects in your images by defining clickable areas. When you click an area in your image that has an Internet address associated with it, you are linked to the specified URL. A clickable area can be a polygon that closely follows an object's shape, a rectangle that matches an object's highlighting box, an oval that fits within an object's highlighting box, or a circle that has a radius equal to the object's longest dimension from its center to its edges.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_URLComment')} **Related Topics**

Layer.URLRegion

Property **URLRegion** As **pntURLRegion**

Description

The **URLRegion** property returns or sets the type of URL region in a layer in Corel PHOTO-PAINT. A URL is a unique address that defines where a Web page is located on the Internet. You can assign Internet addresses, or Uniform Resource Locators (URLs), to the objects in your images by defining clickable areas. When you click an area in your image that has an Internet address associated with it, you are linked to the specified URL. A clickable area can be a polygon that closely follows an object's shape, a rectangle that matches an object's highlighting box, an oval that fits within an object's highlighting box, or a circle that has a radius equal to the object's longest dimension from its center to its edges.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a value of pntURLRegion.

{button ,AL(`CLS_Layer;FNC_URLRegion')} **Related Topics**

pntRegionRectangle	0
pntRegionPolygon	1
pntRegionCircle	2

Layer.Visible

Property **Visible** As Boolean

Description

The **Visible** property returns a True or False value that indicates the visibility status of a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

If the **Visible** property is set the True, the layer is visible in Corel PHOTO-PAINT.

Example

The following code example makes the active layer visible:

```
Sub LayerVisible()
ActiveLayer.Visible = True
End Sub
```

{button ,AL(`CLS_Layer;FNC_Visible')} **Related Topics**

Layer.ZOrder

Property **ZOrder** As Long

ZORDER

Member of **Layer**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Layer;FNC_ZOrder')} **Related Topics**

Layer.Activate

Sub **Activate**()

Description

The **Activate** method opens a layer in Corel PHOTO-PAINT and makes that layer active in the current document. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Example

The following code example activates the active layer:

```
Sub Test()
    ActiveLayer.Activate
End Sub
```

{button ,AL(`CLS_Layer;FNC_Activate')} **Related Topics**

Layer.AffineDistort

Sub **AffineDistort**(ByVal **OffsetX** As Long, ByVal **OffsetY** As Long, ByVal **d11** As Double, ByVal **d12** As Double, ByVal **d21** As Double, ByVal **d22** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **AffineDistort** method distorts a layer in Corel PHOTO-PAINT. Distorting an object lets you stretch it non-proportionately. You can distort an object in the Image Window using the distortion handles.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
OffsetX	Sets the horizontal offset of the distortion.
OffsetY	Sets the vertical offset of the distortion.
d11	Sets a value that creates the distortion of one side of the layer.
d12	Sets a value that creates the distortion of one side of the layer.
d21	Sets a value that creates the distortion of one side of the layer.
d22	Sets a value that creates the distortion of one side of the layer.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Layer;FNC_AffineDistort')} **Related Topics**

Layer.AlignToDocument

Sub **AlignToDocument**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToDocument** method sets the alignment of a layer, according to the alignment type in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_Layer;FNC_AlignToDocument')} **Related Topics**

pntAlignLeft	1
pntAlignRight	2
pntAlignHCenter	3
pntAlignTop	4
pntAlignBottom	8
pntAlignVCenter	12

Layer.AlignToDocumentCenter

Sub **AlignToDocumentCenter**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToDocumentCenter** aligns a layer to the center of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_Layer;FNC_AlignToDocumentCenter')} **Related Topics**

Layer.AlignToGrid

Sub **AlignToGrid**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToGrid** method aligns a layer to the grid in Corel PHOTO-PAINT. A Grid is a series of evenly spaced horizontal and vertical dots that are used to help draw and arrange objects. You can use the controls on the Grid and Ruler Setup dialog box to set the grid's parameters. For greater accuracy, you can also have objects in your illustration snap to the grid when you move, or draw them.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_Layer;FNC_AlignToGrid')} **Related Topics**

Layer.AlignToLayer

Sub **AlignToLayer**(ByVal **Type** As **pntAlignType**, ByVal **Layer** As **Layer**)

Description

The **AlignToLayer** method aligns the active layer to another layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
Layer	The **Layer** parameter identifies the layer to which the active layer is aligned with the **AlignToLayer** method.

{button ,AL(`CLS_Layer;FNC_AlignToLayer')} **Related Topics**

Layer.AlignToLayerRange

Sub **AlignToLayerRange**(ByVal **Type** As **pntAlignType**, ByVal **LayerRange** As **LayerRange**)

Description

The **AlignToLayerRange** method aligns the active layer to a range of layers in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
LayerRange	The **LayerRange** parameter identifies the range of layers to which the active layer is aligned with the **AlignToLayerRange** method.

{button ,AL(`CLS_Layer;FNC_AlignToLayerRange')} **Related Topics**

Layer.AlignToPoint

Sub **AlignToPoint**(ByVal **Type** As **pntAlignType**, ByVal **X** As Long, ByVal **Y** As Long)

Description

The **AlignToPoint** method aligns the active layer to a specified point in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
X	Sets the horizontal coordinate of a point, measured in document units.
Y	Sets the horizontal coordinate of a point, measured in document units.

{button ,AL(`CLS_Layer;FNC_AlignToPoint')} **Related Topics**

Layer.Copy

Function **Copy**() As Boolean

Description

The **Copy** method copies a layer object in Corel PHOTO-PAINT. Copying the object places it in the system clipboard where is can be pasted at a later time. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_Copy')} **Related Topics**

Layer.CreateFloater

Function **CreateFloater**([ByVal Preserve As Boolean = False]) As Boolean

Description

The **CreateFloater** method sets a True or False value that indicates whether or not to create a floating selection in the active layer of Corel PHOTO-PAINT. A floating selection is a selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Preserve	The **Preserve** parameter sets a True or False value that indicates whether or not to maintain the aspect ratio between a floater and the image. This value is optional and the default value is False.

{button ,AL(`CLS_Layer;FNC_CreateFloater')} **Related Topics**

Layer.CreateMask

Sub **CreateMask**([ByVal MaskMode As **pntMaskMode** = pntMaskNormal (0)])

Description

The **CreateMask** method creates a mask on a layer in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
MaskMode	Sets the mask mode in the new mask. This value returns pntMaskMode and is optional. The default value is **pntMaskNormal** (0)

{button ,AL(`CLS_Layer;FNC_CreateMask')} **Related Topics**

Layer.CreatePalette

Function **CreatePalette**(ByVal **Name** As String, [ByVal FileName As String], [ByVal Overwrite As Boolean = False]) As **Palette**

Description

The **CreatePalette** method creates a new color palette from the colors used in a layer in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Name	Sets a string value that names and uniquely identifies the new palette.
FileName	Sets the file name of the new palette. This value is optional.
Overwrite	Sets a True or False value that indicates if the new palette replaces another palette with the same file name. This value is optional and the default value is False.

{button ,AL(`CLS_Layer;FNC_CreatePalette')} **Related Topics**

Layer.CreateSelection

Sub **CreateSelection**()

Description

The **CreateSelection** method creates a selection from the active layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_CreateSelection')} **Related Topics**

Layer.CropToMask

Function **CropToMask**() As Boolean

Description

The **CropToMask** method sets a True or False value that indicates whether or not to crop a layer's image around a mask in Corel PHOTO-PAINT. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_CropToMask')} **Related Topics**

Layer.Cut

Function **Cut**() As Boolean

Description

The **Cut** method cuts a layer object in Corel PHOTO-PAINT. Cutting the object places it in the system clipboard where is can be pasted at a later time. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_Cut')} **Related Topics**

Layer.Delete

Sub **Delete**()

Description

The **Delete** method removes a layer, and its contents, from a document in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Example

The following code example deletes the active layer:

```
Sub Test()
    ActiveLayer.Delete
End Sub
```

{button ,AL(`CLS_Layer;FNC_Delete')} **Related Topics**

pntMaskNormal	0
pntMaskAdd	1
pntMaskSubtract	2
pntMaskXOR	3

Layer.Distort

Sub **Distort**(ByVal **x1** As Long, ByVal **y1** As Long, ByVal **x2** As Long, ByVal **y2** As Long, ByVal **x3** As Long, ByVal **y3** As Long, ByVal **x4** As Long, ByVal **y4** As Long, [ByVal AntiAlias As Boolean = True])

Description

The **Distort** method distorts the shape of the selected layer in Corel PHOTO-PAINT. Distorting an object lets you stretch it non-proportionately. You can distort an object in the Image Window using the distortion handles.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
x1	Sets the distortion value for the horizontal coordinate of a layer's corner.
y1	Sets the distortion value for the vertical coordinate of a layer's corner.
x2	Sets the distortion value for the horizontal coordinate of a layer's corner.
y2	Sets the distortion value for the vertical coordinate of a layer's corner.
x3	Sets the distortion value for the horizontal coordinate of a layer's corner.
y3	Sets the distortion value for the vertical coordinate of a layer's corner.
x4	Sets the distortion value for the horizontal coordinate of a layer's corner.
y4	Sets the distortion value for the vertical coordinate of a layer's corner.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Layer;FNC_Distort')} **Related Topics**

Layer.Duplicate

Function **Duplicate**([ByVal OffsetX As Long = 0], [ByVal OffsetY As Long = 0]) As **Layer**

Description

The **Duplicate** method duplicates the selected layer in Corel PHOTO-PAINT. The new layers are placed in front of existing layers and remain selected. The original layers are deselected.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
OffsetX	The **OffsetX** parameter specifies the horizontal distance to offset the duplicated layer object. This value is optional and the default value is 0.
OffsetY	The **OffsetY** parameter specifies the vertical distance to offset the duplicated layer object. This value is optional and the default value is 0.

{button ,AL(`CLS_Layer;FNC_Duplicate')} **Related Topics**

Layer.Feather

Sub **Feather**(ByVal **Width** As Long, [ByVal Type As **pntFeatherType** = pntFeatherAverage (0)], [ByVal Edges As **pntEdgeType** = pntEdgeLinear (0)])

Description

The **Feather** method feathers the edges of a layer in Corel PHOTO-PAINT. Feathering is the gradual blending of pixels between a selection or an object and the surrounding background. Feathering produces a softer, more natural-looking edge.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Width	Sets the width of a feather effect of a layer's edges, measured in document units.
Type	Sets the feather type, which returns a value of pntFeatherType. This value is optional and the default value is **pntFeatherAverage** (0).
Edges	Sets the feather edge, which returns a value of pntEdgeType. This value is optional and the default value is **pntEdgeLinear** (0)

{button ,AL(`CLS_Layer;FNC_Feather')} **Related Topics**

pntFeatherAverage	0
pntFeatherMiddle	1
pntFeatherOutside	2
pntFeatherInside	3

pntEdgeLinear	0
pntEdgeSquared	1
pntEdgeFlat	2
pntEdgeInverseSquared	3
pntEdgeMesa	4
pntEdgeGaussian	5

Layer.Flip

Sub **Flip**(ByVal **Axes** As **cdrFlipAxes**)

Description

The **Flip** method flips a layer object in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

A layer can be flipped horizontally, vertically, or both.

Parameters	Description
Axes	Sets the manner of the flip in the layer object. This value returns cdrFlipAxes.

{button ,AL(`CLS_Layer;FNC_Flip')} **Related Topics**

cdrFlipHorizontal=1
cdrFlipVertical=2
cdrFlipBoth=3

Layer.GetPosition

Sub **GetPosition**(ByRef **PositionX** As Long, ByRef **PositionY** As Long)

Description

The **GetPosition** method gets the center point of a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
PositionX	Sets the horizontal position of the layer's center point, in document units.
PositionY	Sets the vertical position of the layer's center point, in document units.

{button ,AL(`CLS_Layer;FNC_GetPosition')} **Related Topics**

Layer.GetSize

Sub **GetSize**(ByRef **Width** As Long, ByRef **Height** As Long)

Description

The **GetSize** method gets the width and height of a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Width	The **Width** parameter sets the width of a layer, measured in document units. This value is optional and the default value is 0.
Height	The **Height** parameter sets the height of a layer, measured in document units. This value is optional and the default value is 0.

{button ,AL(`CLS_Layer;FNC_GetSize')} **Related Topics**

Layer.IsInFrontOf

Function **IsInFrontOf**(ByVal **Layer** As **Layer**) As Boolean

Description

The **IsInFrontOf** method returns or sets a True or False value that indicates whether or not a layer is in front of a specified layer in the stacking order of Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

Parameters	Description
Layer	The **Layer** parameter identifies the layer object to see if another layer is in front of this specified layer in the stacking order of Corel PHOTO-PAINT.

{button ,AL(`CLS_Layer;FNC_IsInFrontOf')} **Related Topics**

Layer.Merge

Sub **Merge**()

Description

The **Merge** method merges the selected layer with the background in Corel PHOTO-PAINT. The layer becomes part of the image and cannot be selected again. The manner of a merge depends on the merge mode of the active layer. A merge mode is an editing state that determines how the selected paint, object, or fill color combines with other colors in the image. For example, if you apply color or merge an object into the background using the Normal merge mode, the selected color replaces the original color in the image. If you apply color or merge an object into the background using the Add merge mode, the paint and paper colors are combined to produce a brighter color.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_Merge')} **Related Topics**

Layer.Move

Sub **Move**(ByVal **Dx** As Long, ByVal **Dy** As Long)

Description

The **Move** method changes the position of a layer by moving it a specified distance from its original location in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Dx	The **Dx** parameter specifies the horizontal distance to move a layer with the **Move** method. Delta refers to a limited incremental value in a variable.
Dy	The **Dy** parameter specifies the vertical distance to move a layer with the **Move** method. Delta refers to a limited incremental value in a variable.

{button ,AL(`CLS_Layer;FNC_Move')} **Related Topics**

Layer.OrderBackOf

Sub **OrderBackOf**(ByVal **Layer** As **Layer**)

Description

The **OrderBackOf** method arranges the stacking order of a layer object by moving it in back of a specified layer in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

Parameters	Description
Layer	The **Layer** parameter identifies the layer object that will have another layer placed in back of it in the stacking order of Corel PHOTO-PAINT.

{button ,AL(`CLS_Layer;FNC_OrderBackOf')} **Related Topics**

Layer.OrderBackOne

Sub **OrderBackOne**()

Description

The **OrderBackwardOne** method arranges the stacking order of a layer object by moving it back one in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_Layer;FNC_OrderBackOne')} **Related Topics**

Layer.OrderForwardOne

Sub **OrderForwardOne**()

Description

The **OrderForwardOne** method arranges the stacking order of a layer object by moving it forward one in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_Layer;FNC_OrderForwardOne')} **Related Topics**

Layer.OrderFrontOf

Sub **OrderFrontOf**(ByVal **Layer** As **Layer**)

Description

The **OrderFrontOf** method arranges the stacking order of a layer object by moving it in front of a specified layer in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

Parameters	Description
Layer	The **Layer** parameter identifies the layer object that will have another layer placed in front of it in the stacking order of Corel PHOTO-PAINT.

{button ,AL(`CLS_Layer;FNC_OrderFrontOf')} **Related Topics**

Layer.OrderToBack

Sub **OrderToBack**()

Description

The **OrderToBack** method arranges the stacking order of a layer object by moving it to the back of the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_Layer;FNC_OrderToBack')} **Related Topics**

Layer.OrderToFront

Sub **OrderToFront**()

Description

The **OrderToFront** method arranges the stacking order of a layer object by moving it to the front of the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_Layer;FNC_OrderToFront')} **Related Topics**

Layer.PasteFloater

Function **PasteFloater**([ByVal X As Variant], [ByVal Y As Variant]) As Boolean

Description

The **PasteFloater** method sets a True or False value that indicates whether or not to paste a selection as a floater selection on a layer in Corel PHOTO-PAINT. A floating selection is a selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
X	Sets the horizontal position of the pasted floater selection, measured in document units. This value is optional.
Y	Sets the vertical position of the pasted floater selection, measured in document units. This value is optional.

{button ,AL(`CLS_Layer;FNC_PasteFloater')} **Related Topics**

Layer.Rotate

Sub **Rotate**(ByVal **Angle** As Double, ByVal **CenterX** As Long, ByVal **CenterY** As Long, [ByVal AntiAlias As Boolean = True])

Description

The **Rotate** method rotates a layer by a specified angle around a rotation point in Corel PHOTO-PAINT. Rotating an object means to reposition and reorient the object by turning it around its center of rotation.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Angle	Sets the rotation angle, measured in degrees.
CenterX	Sets the horizontal coordinate of the center or rotation, measured in document units.
CenterY	Sets the vertical coordinate of the center or rotation, measured in document units.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Layer;FNC_Rotate')} **Related Topics**

Layer.SetPosition

Sub **SetPosition**(ByVal **PositionX** As Long, ByVal **PositionY** As Long)

Description

The **SetPosition** method sets the center point of a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
PositionX	Sets the horizontal position of the layer's center point, in document units.
PositionY	Sets the vertical position of the layer's center point, in document units.

{button ,AL(`CLS_Layer;FNC_SetPosition')} **Related Topics**

Layer.SetSize

Sub **SetSize**([ByVal Width As Long = 0], [ByVal Height As Long = 0])

Description

The **SetSize** method sets the width and height of a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Width	The **Width** parameter sets the width of a layer, measured in document units. This value is optional and the default value is 0.
Height	The **Height** parameter sets the height of a layer, measured in document units. This value is optional and the default value is 0.

{button ,AL(`CLS_Layer;FNC_SetSize')} **Related Topics**

Layer.Skew

Sub **Skew**(ByVal **HAngle** As Double, ByVal **VAngle** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **Skew** method skews a layer in Corel PHOTO-PAINT. Skewing refers to a slant of an object. Skewing a layer slants the layer in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
HAngle	The **HAngle** parameter specifies the degree in which to slant the layer horizontally.
VAngle	The **VAngle** parameter specifies the degree in which to slant the layer vertically.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Layer;FNC_Skew')} **Related Topics**

Layer.Stretch

Sub **Stretch**(ByVal **XOrigin** As Long, ByVal **YOrigin** As Long, ByVal **XScale** As Double, ByVal **YScale** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **Stretch** method stretches a layer in Corel PHOTO-PAINT. Stretching an object means to size an object horizontally or vertically. Stretching changes the size of an object in one direction only, as opposed to sizing, which maintains the aspect ratio (the ratio of height to width).

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
XOrigin	The **XOrigin** parameter specifies the horizontal position of the anchor point when stretching a layer. The anchor point is the point that remains stationary when you stretch, scale, mirror, or skew an object. This value is optional and the default value is 0. This value is measured in document units.
YOrigin	The **YOrigin** parameter specifies the vertical position of the anchor point when stretching a layer. The anchor point is the point that remains stationary when you stretch, scale, mirror, or skew an object. This value is optional and the default value is 0. This value is measured in document units.
XScale	The **XScale** parameter specifies the width value of the aspect ratio used when stretching objects. The aspect ratio is the ratio of the width of an image to its height (expressed mathematically as x:y). For example, the aspect ratio of an image that is 640 x 480 pixels is 4:3. This value is optional and the default value is 100.
YScale	The **YScale** parameter specifies the width value of the aspect ratio used when stretching objects. The aspect ratio is the ratio of the width of an image to its height (expressed mathematically as x:y). For example, the aspect ratio of an image that is 640 x 480 pixels is 4:3. This value is optional and the default value is 100.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Layer;FNC_Stretch')} **Related Topics**

Layer.Threshold

Sub **Threshold**(ByVal **Threshold** As Long)

Description

The **Threshold** method alters the edges of mask selections or objects by removing the smooth transition between a selection, or a layer, and the underlying image in Corel PHOTO-PAINT. Threshold is a level of tolerance for tonal variation in a bitmap image. For example, when you convert an image to the Black-and-White color mode, the threshold you set determines how many tonal values are converted to black and how many to white. Threshold settings are also used in color-sensitive masks and some Effects filters.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Threshold	The **Threshold** parameter lets you change threshold values for the colors in an image by converting a range of colors to black or white. The threshold values that you set determine which pixels become black and which pixels become white. If you convert colors to black, only pixels with a brightness value that is lower than the threshold value are converted to black. If you convert colors to white, only pixels with a brightness value that is higher than the threshold value are converted to white. The Threshold brightness values range from 0 (black) to 255 (white).

{button ,AL(`CLS_Layer;FNC_Threshold')} **Related Topics**

Layer.Ungroup

Sub **Ungroup**()

Description

The **Ungroup** method breaks up the selected group into its individual objects in a layer in Corel PHOTO-PAINT. Ungrouping is a command that causes a set of objects acting as one unit to behave as individual objects. If you have more than one sublevel of grouping, Ungroup breaks up one level of grouping at a time. A group is a set of objects that behave as one unit. Also refers to the Group command. Most operations you perform on a group apply equally to each of its components.

Grouping objects lets you move, delete, or transform objects as a single entity. When you group objects, a highlighting box appears around them in the Image Window and a thick, black line connects the objects in the Objects Docker window.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_Ungroup')} **Related Topics**

Layer.UngroupAll

Sub **UngroupAll**()

Description

The **UngroupAll** method removes the grouping feature from all nested groups of layer objects in Corel PHOTO-PAINT. A group is a set of objects that behave as one unit. Also refers to the Group command. Most operations you perform on a group apply equally to each of its components.

Grouping objects lets you move, delete, or transform objects as a single entity. When you group objects, a highlighting box appears around them in the Image Window and a thick, black line connects the objects in the Objects Docker window.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layer;FNC_UngroupAll')} **Related Topics**

LayerRange properties

LayerRange Legend

- Application

- Count

-
- Item

- Parent
  - PositionX
  - PositionY

- SizeHeight

- SizeWidth

LayerRange methods

LayerRange Legend

- Add
- AddClipMask
- AddRange
- AddToSelection
- AffineDistort
- AlignToDocument
- AlignToDocumentCenter
- AlignToGrid
- AlignToLayer
- AlignToLayerRange
- AlignToPoint
- Combine
- Copy
- CreateMask
- CreatePalette
- CreateSelection
- CropToMask
- Cut
- Defringe
- Delete
- Distort
- Distribute
- DistributeSpace
- Duplicate
- Feather
- Flip
- GetPosition
- GetSize
- Group
- Merge
- Move
- OrderBackOf
- OrderBackOne
- OrderForwardOne
- OrderFrontOf
- OrderReverse
- OrderToBack
- OrderToFront
- Remove
- RemoveAll
- RemoveFromSelection
- RemoveMatte
- Rotate
- SetMergeMode
- SetOpacity
- SetOrder
- SetPosition
- SetSize
- Skew
- Stretch
- Threshold

UngroupAll

LayerRange

Class **LayerRange**

Properties Methods Referenced By

The **LayerRange** class defines the characteristics of a range of layer objects and describes the look and behavior of the objects through its properties and methods.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

The layering feature of Corel PHOTO-PAINT gives you added flexibility for organizing and editing the objects in your images. You can divide an image into multiple layers, each containing a portion of the image's contents. Together, layers act as a hierarchy that helps determine the vertical arrangement of an image's components. In this arrangement (the stacking order), objects on the top layer always overlay objects on the layer below.

{button ,AL(`CLS_LayerRange')} **Related Topics**

LayerRange.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_LayerRange;FNC_Application')} **Related Topics**

LayerRange.Count

Property **Count** As Long

Description

The **Count** property returns the number of layers in a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_LayerRange;FNC_Count')} **Related Topics**

LayerRange.Item

Property **Item**(ByVal **IndexOrName** As Variant) As **Layer**

Description

The **Item** property returns a value associated with the index number of a layer object in a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the range. You must reference a layer in the collection by passing its index number as a parameter:

Parameters	Description
IndexOrName	**Index** is a preset placeholder for each object in a Layer Range; it uniquely identifies each member of the range. **Name** is the unique string value that identifies each layer.

{button ,AL(`CLS_LayerRange;FNC_Item')} **Related Topics**

LayerRange.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_LayerRange;FNC_Parent')} **Related Topics**

LayerRange.PositionX

Property **PositionX** As Long

Description

The **PositionX** property returns or set the horizontal position of the center of a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_PositionX')} **Related Topics**

LayerRange.PositionY

Property **PositionY** As Long

Description

The **PositionY** property returns or set the vertical position of the center of a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_PositionY')} **Related Topics**

LayerRange.SizeHeight

Property **SizeHeight** As Long

Description

The **SizeHeight** property returns the height of a layer in Corel PHOTO-PAINT, measured in documents units. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_LayerRange;FNC_SizeHeight')} **Related Topics**

LayerRange.SizeWidth

Property **SizeWidth** As Long

Description

The **SizeWidth** property returns or sets the width of a layer in Corel PHOTO-PAINT, measured in documents units. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_LayerRange;FNC_SizeWidth')} **Related Topics**

LayerRange.Add

Sub **Add**(ByVal **Layer** As **Layer**)

Description

The **Add** method adds a layer to the layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Layer	The **Layer** parameter identifies the layer to which is added to the layer range with the **Add** method.

{button ,AL(`CLS_LayerRange;FNC_Add')} **Related Topics**

LayerRange.AddClipMask

Sub **AddClipMask**(ByVal **CreationMode** As **pntClipMaskMode**, [ByVal OpacityLevel As Long = 0])

Description

The **AddClipMask** method adds a clip mask object to a layer rangein Corel PHOTO-PAINT. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
CreationMode	Sets the creation mode for the new clip mask on the object layer. This value returns pntClipMaskMode.
OpacityLevel	Sets the level of opacity in the new clip mask on the object layer. This value is optional and the default value is 0. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects.

{button ,AL(`CLS_LayerRange;FNC_AddClipMask')} **Related Topics**

LayerRange.AddRange

Sub **AddRange**(ByVal **LayerRange** As **LayerRange**)

Description

The **AddRange** method adds a layer rangeto an existing layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
LayerRange	The LayerRange parameter is a range of index numbers that identify and group specific layers. By selecting a range of index numbers, you are selecting specific layer objects that become members of the array.

{button ,AL(`CLS_LayerRange;FNC_AddRange')} **Related Topics**

LayerRange.AddToSelection

Sub **AddToSelection**()

Description

The **AddToSelection** method adds the layer rangefrom the current selection in Corel PHOTO-PAINT. A selection is an area of an image that is not protected by a mask and that is, therefore, available for editing. The selection is affected by the use of painting and editing tools, special effects, and image commands.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_AddToSelection')} **Related Topics**

LayerRange.AffineDistort

Sub **AffineDistort**(ByVal **OffsetX** As Long, ByVal **OffsetY** As Long, ByVal **d11** As Double, ByVal **d12** As Double, ByVal **d21** As Double, ByVal **d22** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **AffineDistort** method distorts a layer rangein Corel PHOTO-PAINT. Distorting an object lets you stretch it non-proportionately. You can distort an object in the Image Window using the distortion handles.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
OffsetX	Sets the horizontal offset of the distortion.
OffsetY	Sets the vertical offset of the distortion.
d11	Sets a value that creates the distortion of one side of the layer.
d12	Sets a value that creates the distortion of one side of the layer.
d21	Sets a value that creates the distortion of one side of the layer.
d22	Sets a value that creates the distortion of one side of the layer.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_LayerRange;FNC_AffineDistort')} **Related Topics**

LayerRange.AlignToDocument

Sub **AlignToDocument**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToDocument** method sets the alignment of a layer range, according to the alignment type in a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_LayerRange;FNC_AlignToDocument')} **Related Topics**

LayerRange.AlignToDocumentCenter

Sub **AlignToDocumentCenter**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToDocumentCenter** aligns a layer rangeto the center of a document in Corel PHOTO-PAINT. A document in Corel PHOTO-PAINT is single page image or a collection of single-page images that make up a multi-page document.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_LayerRange;FNC_AlignToDocumentCenter')} **Related Topics**

LayerRange.AlignToGrid

Sub **AlignToGrid**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToGrid** method aligns a layer rangeto the grid in Corel PHOTO-PAINT. A Grid is a series of evenly spaced horizontal and vertical dots that are used to help draw and arrange objects. You can use the controls on the Grid and Ruler Setup dialog box to set the grid's parameters. For greater accuracy, you can also have objects in your illustration snap to the grid when you move, or draw them.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_LayerRange;FNC_AlignToGrid')} **Related Topics**

LayerRange.AlignToLayer

Sub **AlignToLayer**(ByVal **Type** As **pntAlignType**, ByVal **Layer** As **Layer**)

Description

The **AlignToLayer** method aligns the layer range to another layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
Layer	The **Layer** parameter identifies the layer to which the active layer is aligned with the **AlignToLayer** method.

{button ,AL(`CLS_LayerRange;FNC_AlignToLayer')} **Related Topics**

LayerRange.AlignToLayerRange

Sub **AlignToLayerRange**(ByVal **Type** As **pntAlignType**, ByVal **LayerRange** As **LayerRange**)

Description

The **AlignToLayerRange** method aligns the active layer to a range of layers in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
LayerRange	The **LayerRange** parameter identifies the range of layers to which the active layer is aligned with the **AlignToLayerRange** method.

{button ,AL(`CLS_LayerRange;FNC_AlignToLayerRange')} **Related Topics**

LayerRange.AlignToPoint

Sub **AlignToPoint**(ByVal **Type** As **pntAlignType**, ByVal **X** As Long, ByVal **Y** As Long)

Description

The **AlignToPoint** method aligns the layer range to a specified point in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
X	Sets the horizontal coordinate of a point, measured in document units.
Y	Sets the horizontal coordinate of a point, measured in document units.

{button ,AL(`CLS_LayerRange;FNC_AlignToPoint')} **Related Topics**

LayerRange.Combine

Function **Combine**() As **Layer**

Description

The **Combine** method combines the layers of a layer rangewith an object in Corel PHOTO-PAINT. When you combine objects, you permanently create one object from multiple objects. You can also combine objects with the background to decrease the file size of an image. When you combine objects with the background, they no longer float above the rest of the image, and they cannot be selected or changed individually.

{button ,AL(`CLS_LayerRange;FNC_Combine')} **Related Topics**

LayerRange.Copy

Function **Copy**() As Boolean

Description

The **Copy** method returns a True or False value that indicates whether or not to copy a layer rangein Corel PHOTO-PAINT. Copying the object places it in the system clipboard where is can be pasted at a later time. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_Copy')} **Related Topics**

LayerRange.CreateMask

Sub **CreateMask**([ByVal MaskMode As **pntMaskMode** = pntMaskNormal (0)])

Description

The **CreateMask** method creates a mask on a range of layers in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
MaskMode	Sets the mask mode in the new mask. This value returns pntMaskMode and is optional. The default value is **pntMaskNormal** (0)

{button ,AL(`CLS_LayerRange;FNC_CreateMask')} **Related Topics**

LayerRange.CreatePalette

Function **CreatePalette**(ByVal **Name** As String, [ByVal FileName As String], [ByVal Overwrite As Boolean = False]) As **Palette**

Description

The **CreatePalette** method creates a new color palette from the colors used in a layer rangein Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Name	Sets a string value that names and uniquely identifies the new palette.
FileName	Sets the file name of the new palette. This value is optional.
Overwrite	Sets a True or False value that indicates if the new palette replaces another palette with the same file name. This value is optional and the default value is False.

{button ,AL(`CLS_LayerRange;FNC_CreatePalette')} **Related Topics**

LayerRange.CreateSelection

Sub **CreateSelection**()

Description

The **CreateSelection** method creates a selection from the layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_CreateSelection')} **Related Topics**

LayerRange.CropToMask

Sub **CropToMask**()

Description

The **CropToMask** method sets a True or False value that indicates whether or not to crop a layer's image, in a layer range around a mask in Corel PHOTO-PAINT. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_CropToMask')} **Related Topics**

LayerRange.Cut

Function **Cut**() As Boolean

Description

The **Cut** method sets a True or False value that indicates whether of not to cut a layer rangein Corel PHOTO-PAINT. Cutting the object places it in the system clipboard where is can be pasted at a later time. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_Cut')} **Related Topics**

LayerRange.Defringe

Sub **Defringe**(ByVal **Amount** As Long)

Description

The **Defringe** method defringes the edges of the layers in a layer rangein Corel PHOTO-PAINT. The **Defringe** method replaces the color of the pixels that lie along the outside edge of an object with the color of the pixels that lie inside the boundaries of the object.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

An object can have stray pixels along its edges; these strays contrast with neighboring pixels, giving the object a slightly ragged effect. Defringing stray pixels blends the object with the background to create a smooth, polished appearance.

Parameters	Description
Amount	Sets the level of the defringe effect in the edges of the object layer. Large defringe values create a gradual transition between the edges of the object and the background. Values range from 1 to 100.

{button ,AL(`CLS_LayerRange;FNC_Defringe')} **Related Topics**

LayerRange.Delete

Sub **Delete**()

Description

The **Delete** method removes a layer range, and its contents, from a document in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_Delete')} **Related Topics**

LayerRange.Distort

Sub **Distort**(ByVal **x1** As Long, ByVal **y1** As Long, ByVal **x2** As Long, ByVal **y2** As Long, ByVal **x3** As Long, ByVal **y3** As Long, ByVal **x4** As Long, ByVal **y4** As Long, [ByVal AntiAlias As Boolean = True])

Description

The **Distort** method distorts the shape of the layer range in Corel PHOTO-PAINT. Distorting an object lets you stretch it non-proportionately. You can distort an object in the Image Window using the distortion handles.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
x1	Sets the distortion value for the horizontal coordinate of a layer's corner.
y1	Sets the distortion value for the vertical coordinate of a layer's corner.
x2	Sets the distortion value for the horizontal coordinate of a layer's corner.
y2	Sets the distortion value for the vertical coordinate of a layer's corner.
x3	Sets the distortion value for the horizontal coordinate of a layer's corner.
y3	Sets the distortion value for the vertical coordinate of a layer's corner.
x4	Sets the distortion value for the horizontal coordinate of a layer's corner.
y4	Sets the distortion value for the vertical coordinate of a layer's corner.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_LayerRange;FNC_Distort')} **Related Topics**

LayerRange.Distribute

Sub **Distribute**(ByVal **Type** As **pntDistributeType**, [ByVal DocumentExtent As Boolean = False])

DISTRIBUTE

Member of **LayerRange**

Parameters	Description
Type	Description of Type goes here (**in**)
DocumentExtent	Description of DocumentExtent goes here (**in**) Optional Default value = **False**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_LayerRange;FNC_Distribute')} **Related Topics**

LayerRange.DistributeSpace

Sub **DistributeSpace**(ByVal **Type** As **pntDistributeType**, [ByVal SpaceX As Long = 0], [ByVal SpaceY As Long = 0])

DISTRIBUTESPACE

Member of **LayerRange**

Parameters	Description
Type	Description of Type goes here (**in**)
SpaceX	Description of SpaceX goes here (**in**) Optional Default value = **0**
SpaceY	Description of SpaceY goes here (**in**) Optional Default value = **0**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_LayerRange;FNC_DistributeSpace')} **Related Topics**

LayerRange.Duplicate

Function **Duplicate**([ByVal OffsetX As Long = 0], [ByVal OffsetY As Long = 0]) As **LayerRange**

Description

The **Duplicate** method duplicates the layer range in Corel PHOTO-PAINT. The new layers are placed in front of existing layers and remain selected. The original layers are deselected.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
OffsetX	The **OffsetX** parameter specifies the horizontal distance to offset the duplicated layer object. This value is optional and the default value is 0.
OffsetY	The **OffsetY** parameter specifies the vertical distance to offset the duplicated layer object. This value is optional and the default value is 0.

{button ,AL(`CLS_LayerRange;FNC_Duplicate')} **Related Topics**

LayerRange.Feather

Sub **Feather**(ByVal **Width** As Long, [ByVal Type As **pntFeatherType** = pntFeatherAverage (0)], [ByVal Edges As **pntEdgeType** = pntEdgeLinear (0)])

Description

The **Feather** method feathers the edges of a layer range in Corel PHOTO-PAINT. Feathering is the gradual blending of pixels between a selection or an object and the surrounding background. Feathering produces a softer, more natural-looking edge.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Width	Sets the width of a feather effect of a layer's edges, measured in document units.
Type	Sets the feather type, which returns a value of pntFeatherType. This value is optional and the default value is **pntFeatherAverage** (0).
Edges	Sets the feather edge, which returns a value of pntEdgeType. This value is optional and the default value is **pntEdgeLinear** (0)

{button ,AL(`CLS_LayerRange;FNC_Feather')} **Related Topics**

LayerRange.Flip

Sub **Flip**(ByVal **Axes** As **cdrFlipAxes**)

Description

The **Flip** method flips a layer rangein Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

A layer can be flipped horizontally, vertically, or both.

Parameters	Description
Axes	Sets the manner of the flip in the layer object. This value returns cdrFlipAxes.

{button ,AL(`CLS_LayerRange;FNC_Flip')} **Related Topics**

LayerRange.GetPosition

Sub **GetPosition**(ByRef **PositionX** As Long, ByRef **PositionY** As Long)

Description

The **GetPosition** method gets the center point of a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
PositionX	Sets the horizontal position of the layer's center point, in document units.
PositionY	Sets the vertical position of the layer's center point, in document units.

{button ,AL(`CLS_LayerRange;FNC_GetPosition')} **Related Topics**

LayerRange.GetSize

Sub **GetSize**(ByRef **Width** As Long, ByRef **Height** As Long)

Description

The **GetSize** method gets the width and height of a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Width	The **Width** parameter sets the width of a layer, measured in document units. This value is optional and the default value is 0.
Height	The **Height** parameter sets the height of a layer, measured in document units. This value is optional and the default value is 0.

{button ,AL(`CLS_LayerRange;FNC_GetSize')} **Related Topics**

LayerRange.Group

Sub **Group**()

Description

The **Group** method groups a layer rangein Corel PHOTO-PAINT. Grouping objects lets you move, delete, or transform objects as a single entity. When you group objects, a highlighting box appears around them in the Image Window and a thick, black line connects the objects in the Objects Docker window.

{button ,AL(`CLS_LayerRange;FNC_Group')} **Related Topics**

LayerRange.Merge

Sub **Merge**()

Description

The **Merge** method merges the layer range with the background in Corel PHOTO-PAINT. The layer becomes part of the image and cannot be selected again. The manner of a merge depends on the merge mode of the active layer. A merge mode is an editing state that determines how the selected paint, object, or fill color combines with other colors in the image. For example, if you apply color or merge an object into the background using the Normal merge mode, the selected color replaces the original color in the image. If you apply color or merge an object into the background using the Add merge mode, the paint and paper colors are combined to produce a brighter color.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_Merge')} **Related Topics**

LayerRange.Move

Sub **Move**(ByVal **Dx** As Long, ByVal **Dy** As Long)

Description

The **Move** method changes the position of a layer range by moving it a specified distance from its original location in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Dx	The **Dx** parameter specifies the horizontal distance to move a layer with the **Move** method. Delta refers to a limited incremental value in a variable.
Dy	The **Dy** parameter specifies the vertical distance to move a layer with the **Move** method. Delta refers to a limited incremental value in a variable.

{button ,AL(`CLS_LayerRange;FNC_Move')} **Related Topics**

LayerRange.OrderBackOf

Sub **OrderBackOf**(ByVal **Layer** As **Layer**)

Description

The **OrderBackOf** method arranges the stacking order of a layer rangeby moving it in back of a specified layer in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

Parameters	Description
Layer	The **Layer** parameter identifies the layer object that will have another layer placed in back of it in the stacking order of Corel PHOTO-PAINT.

{button ,AL(`CLS_LayerRange;FNC_OrderBackOf')} **Related Topics**

LayerRange.OrderBackOne

Sub **OrderBackOne**()

Description

The **OrderBackwardOne** method arranges the stacking order of a layer rangeby moving it back one in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_LayerRange;FNC_OrderBackOne')} **Related Topics**

LayerRange.OrderForwardOne

Sub **OrderForwardOne**()

Description

The **OrderForwardOne** method arranges the stacking order of a layer rangeby moving it forward one in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_LayerRange;FNC_OrderForwardOne')} **Related Topics**

LayerRange.OrderFrontOf

Sub **OrderFrontOf**(ByVal **Layer** As **Layer**)

Description

The **OrderFrontOf** method arranges the stacking order of a layer rangeby moving it in front of a specified layer in the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

Parameters	Description
Layer	The **Layer** parameter identifies the layer object that will have another layer placed in front of it in the stacking order of Corel PHOTO-PAINT.

{button ,AL(`CLS_LayerRange;FNC_OrderFrontOf')} **Related Topics**

LayerRange.OrderReverse

Sub **OrderReverse**()

Description

The **OrderReverse** method reverses the stacking order of layers in a layer rangein Corel PHOTO-PAINT. Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_OrderReverse')} **Related Topics**

LayerRange.OrderToBack

Sub **OrderToBack**()

Description

The **OrderToBack** method arranges the stacking order of a layer rangeby moving it to the back of the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_LayerRange;FNC_OrderToBack')} **Related Topics**

LayerRange.OrderToFront

Sub **OrderToFront**()

Description

The **OrderToFront** method arranges the stacking order of a layer rangeby moving it to the front of the stacking order in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Stacking order is the sequence in which objects are created in the Image Window. This order determines the relationship between objects and, therefore, the appearance of your image. The first object you create appears on the bottom; the last object appears on the top. You can use the Order commands to place the objects where you want them; however, the background object always appears on the bottom and cannot be reordered.

{button ,AL(`CLS_LayerRange;FNC_OrderToFront')} **Related Topics**

LayerRange.Remove

Sub **Remove**(ByVal **Index** As Long)

Description

The **Remove** method removes a layer from a layer range in Corel PHOTO-PAINT, by referencing the index of the layer. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Index	The **Index** parameter specifies the index number of the layer in the layer range that is removed with the **Remove** method.

{button ,AL(`CLS_LayerRange;FNC_Remove')} **Related Topics**

LayerRange.RemoveAll

Sub **RemoveAll**()

Description

The **RemoveAll** method removes all layers from a layer rangein Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_RemoveAll')} **Related Topics**

LayerRange.RemoveFromSelection

Sub **RemoveFromSelection**()

Description

The **RemoveFromSelection** method removes the layer rangefrom the current selection in Corel PHOTO-PAINT. A selection is an area of an image that is not protected by a mask and that is, therefore, available for editing. The selection is affected by the use of painting and editing tools, special effects, and image commands.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_RemoveFromSelection')} **Related Topics**

LayerRange.RemoveMatte

Sub **RemoveMatte**(ByVal **Type** As **pntMatteType**)

Description

The **RemoveMatte** method removes a matte from the layer range in Corel PHOTO-PAINT. You can change the transparency of the pixels along the edge of a feathered object using the Remove Matte commands. The more transparent the pixels are, the more you can see through them. The Remove Black Matte command makes semitransparent pixels more transparent. The Remove White Matte command makes semitransparent pixels less transparent.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	Sets the type of matte that is removed from an object layer. This value returns pntMatteType.

{button ,AL(`CLS_LayerRange;FNC_RemoveMatte')} **Related Topics**

LayerRange.Rotate

Sub **Rotate**(ByVal **Angle** As Double, ByVal **CenterX** As Long, ByVal **CenterY** As Long, [ByVal AntiAlias As Boolean = True])

Description

The **Rotate** method rotates a layer range by a specified angle around a rotation point in Corel PHOTO-PAINT. Rotating an object means to reposition and reorient the object by turning it around its center of rotation.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Angle	Sets the rotation angle, measured in degrees.
CenterX	Sets the horizontal coordinate of the center or rotation, measured in document units.
CenterY	Sets the vertical coordinate of the center or rotation, measured in document units.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_LayerRange;FNC_Rotate')} **Related Topics**

LayerRange.SetMergeMode

Sub **SetMergeMode**(ByVal **MergeMode** As **pntMergeMode**)

Description

The **SetMergeMode** method sets the merge mode in a layer rangein Corel PHOTO-PAINT. A merge mode is an editing state that determines how the selected paint, object, or fill color combines with other colors in the image. For example, if you apply color or merge an object into the background using the Normal merge mode, the selected color replaces the original color in the image. If you apply color or merge an object into the background using the Add merge mode, the paint and paper colors are combined to produce a brighter color.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
MergeMode	The **MergeMode** parameter sets the merge mode of the new layer range. This value returns pntMergeMode. This value is optional and the default value is pntMergeNormal (0)

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_LayerRange;FNC_SetMergeMode')} **Related Topics**

LayerRange.SetOpacity

Sub **SetOpacity**(ByVal **Opacity** As Long)

Description

The **SetOpacity** method sets the level of opacity in a layer rangein Corel PHOTO-PAINT. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Opacity	The **Opacity** parameter sets the opacity level of the new layer range. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects. This value is optional and the default value is 100.

{button ,AL(`CLS_LayerRange;FNC_SetOpacity')} **Related Topics**

LayerRange.SetOrder

Sub **SetOrder**(ByVal **Order** As Long)

Description

The **SetOrder** method sets the order of layers in a layer rangein Corel PHOTO-PAINT. When you create multiple objects in an image, they are stacked on top of one another in the order in which they are created. The most recently created object is at the top of the stack, and the image background is always at the bottom. This stacking order determines the relationship between objects according to the sequence in which they are drawn. For example, you can move an object in the Image Window to cover an object that is lower in the stacking order. You can change the order of objects in the stack using the Order commands in the Object menu or using the Objects Docker window.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Order	Sets the stacking order of layers in a layer range.

{button ,AL(`CLS_LayerRange;FNC_SetOrder')} **Related Topics**

LayerRange.SetPosition

Sub **SetPosition**(ByVal **PositionX** As Long, ByVal **PositionY** As Long)

Description

The **SetPosition** method sets the center point of a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
PositionX	Sets the horizontal position of the layer's center point, in document units.
PositionY	Sets the vertical position of the layer's center point, in document units.

{button ,AL(`CLS_LayerRange;FNC_SetPosition')} **Related Topics**

LayerRange.SetSize

Sub **SetSize**([ByVal Width As Long = 0], [ByVal Height As Long = 0])

Description

The **SetSize** method sets the width and height of a layer range in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Width	The **Width** parameter sets the width of a layer, measured in document units. This value is optional and the default value is 0.
Height	The **Height** parameter sets the height of a layer, measured in document units. This value is optional and the default value is 0.

{button ,AL(`CLS_LayerRange;FNC_SetSize')} **Related Topics**

LayerRange.Skew

Sub **Skew**(ByVal **HAngle** As Double, ByVal **VAngle** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **Skew** method skews a layer rangein Corel PHOTO-PAINT. Skewing refers to a slant of an object. Skewing a layer slants the layer in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
HAngle	The **HAngle** parameter specifies the degree in which to slant the layer horizontally.
VAngle	The **VAngle** parameter specifies the degree in which to slant the layer vertically.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_LayerRange;FNC_Skew')} **Related Topics**

LayerRange.Stretch

Sub **Stretch**(ByVal **XOrigin** As Long, ByVal **YOrigin** As Long, ByVal **XScale** As Double, ByVal **YScale** As Double, [ByVal AntiAlias As Boolean = True], [ByVal Relative As Boolean = False])

Description

The **Stretch** method stretches a layer range in Corel PHOTO-PAINT. Stretching an object means to size an object horizontally or vertically. Stretching changes the size of an object in one direction only, as opposed to sizing, which maintains the aspect ratio (the ratio of height to width).

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
XOrigin	The **XOrigin** parameter specifies the horizontal position of the anchor point when stretching a layer. The anchor point is the point that remains stationary when you stretch, scale, mirror, or skew an object. This value is optional and the default value is 0. This value is measured in document units.
YOrigin	The **YOrigin** parameter specifies the vertical position of the anchor point when stretching a layer. The anchor point is the point that remains stationary when you stretch, scale, mirror, or skew an object. This value is optional and the default value is 0. This value is measured in document units.
XScale	The **XScale** parameter specifies the width value of the aspect ratio used when stretching objects. The aspect ratio is the ratio of the width of an image to its height (expressed mathematically as x:y). For example, the aspect ratio of an image that is 640 x 480 pixels is 4:3. This value is optional and the default value is 100.
YScale	The **YScale** parameter specifies the width value of the aspect ratio used when stretching objects. The aspect ratio is the ratio of the width of an image to its height (expressed mathematically as x:y). For example, the aspect ratio of an image that is 640 x 480 pixels is 4:3. This value is optional and the default value is 100.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_LayerRange;FNC_Stretch')} **Related Topics**

LayerRange.Threshold

Sub **Threshold**(ByVal **Threshold** As Long)

Description

The **Threshold** method alters the edges of mask selections or objects by removing the smooth transition between a selection, or a layer range, and the underlying image in Corel PHOTO-PAINT. Threshold is a level of tolerance for tonal variation in a bitmap image. For example, when you convert an image to the Black-and-White color mode, the threshold you set determines how many tonal values are converted to black and how many to white. Threshold settings are also used in color-sensitive masks and some Effects filters.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Threshold	The **Threshold** parameter lets you change threshold values for the colors in an image by converting a range of colors to black or white. The threshold values that you set determine which pixels become black and which pixels become white. If you convert colors to black, only pixels with a brightness value that is lower than the threshold value are converted to black. If you convert colors to white, only pixels with a brightness value that is higher than the threshold value are converted to white. The Threshold brightness values range from 0 (black) to 255 (white).

{button ,AL(`CLS_LayerRange;FNC_Threshold')} **Related Topics**

LayerRange.UngroupAll

Sub **UngroupAll**()

Description

The **UngroupAll** method removes the grouping feature from all nested groups of a layer range in Corel PHOTO-PAINT. A group is a set of objects that behave as one unit. Also refers to the Group command. Most operations you perform on a group apply equally to each of its components.

Grouping objects lets you move, delete, or transform objects as a single entity. When you group objects, a highlighting box appears around them in the Image Window and a thick, black line connects the objects in the Objects Docker window.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_LayerRange;FNC_UngroupAll')} **Related Topics**

Layers properties

Layers Legend

- Application
- Count
-
- Item
- Parent

Layers methods

Layers Legend

- Add
- All
- AllExcluding
- Merge
- Paste
- Range
- Selection

A range is series of similar objects in Corel PHOTO-PAINT.

Layers

Class **Layers**

Properties Methods Referenced By

The **Layers** class defines the characteristics of **Layers** collection objects and describes the look and behavior of the objects through its properties and methods.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

The layering feature of Corel PHOTO-PAINT gives you added flexibility for organizing and editing the objects in your images. You can divide an image into multiple layers, each containing a portion of the image's contents. Together, layers act as a hierarchy that helps determine the vertical arrangement of an image's components. In this arrangement (the stacking order), objects on the top layer always overlay objects on the layer below.

{button ,AL(`CLS_Layers')} **Related Topics**

Layers.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Layers;FNC_Application')} **Related Topics**

Layers.Count

Property **Count** As Long

Description

The **Count** property returns the number of layers in the **Layers** collection of Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain.

The **Count** property returns a Read-Only value.

{button ,AL(`CLS_Layers;FNC_Count')} **Related Topics**

Layers.Item

Property **Item**(ByVal **IndexOrName** As Variant) As **Layer**

Description

The **Item** property returns a value associated with the index number of a layer object in the **Layers** collection of Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain.

`Layers(2)` refers to the second layer object in the **Layers** collection of Corel PHOTO-PAINT.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Layers.Item(2)` is the same as `Layers(2)` - they both reference the second layer object in the **Layers** collection.

You must reference a layer in the collection by passing its index number as a parameter:

Parameters	Description
IndexOrName	**Index** is a preset placeholder for each object in a **Layers** collection; it uniquely identifies each member of the collection. **Name** is the unique string value that identifies each layer.

{button ,AL(`CLS_Layers;FNC_Item')} **Related Topics**

Layers.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Layers;FNC_Parent')} **Related Topics**

Layers.Add

Function **Add**([ByVal Name As String], [ByVal Opacity As Long = 100], [ByVal MergeMode As **pntMergeMode** = pntMergeNormal (0)], [ByVal CreationMode As **pntCreationMode** = pntEmptyLayer (0)], [ByVal Source As Object = Nothing]) As **Layer**

Description

The **Add** method adds a layer to the **Layers** collection in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Name	The **Name** parameter sets the name of the new layer in the collection. This value is optional.
Opacity	The **Opacity** parameter sets the opacity level of the new layer in the collection. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects. This value is optional and the default value is 100.
MergeMode	The **MergeMode** parameter sets the merge mode of the new layer in the collection. This value returns pntMergeMode. This value is optional and the default value is **pntMergeNormal** (0)
CreationMode	The **CreationMode** parameter sets the creation mode of the new layer in the collection. This value returns pntCreationMode. This value is optional and the default value is **pntEmptyLayer** (0)
Source	The **Source** parameter sets the source of the new layer in the collection. This value is optional and the new value is **Nothing.**

{button ,AL(`CLS_Layers;FNC_Add')} **Related Topics**

pntEmptyLayer	0
pntFromBackground	1
pntCopySelection	2
pntCutSelection	3

pntMergeNormal	0
pntMergeAdd	1
pntMergeSubtract	2
pntMergeDifference	3
pntMergeMultiply	4
pntMergeDivide	5
pntMergeIfLighter	6
pntMergeIfDarker	7
pntMergeTexturize	8
pntMergeInvert	9
pntMergeAND	10
pntMergeOR	11
pntMergeXOR	12
pntMergeBehind	13
pntMergeScreen	14
pntMergeOverlay	15
pntMergeSoftLight	16
pntMergeHardLight	17
pntMergeColorDodge	18
pntMergeColorBurn	19
pntMergeExclusion	20

Layers.All

Function **All**() As **LayerRange**

Description

The **All** method returns or sets a value associated with all layer objects in the **Layers** collection in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

The **All** method allows you to reference all layers concurrently. The use of the **All** method makes it easier and faster to find, sort, and edit all layers in the Layers collection.

{button ,AL(`CLS_Layers;FNC_All')} **Related Topics**

Layers.AllExcluding

Function **AllExcluding**(ByRef **IndexArray**() As Variant) As **LayerRange**

Description

The **AllExcluding** method returns or sets a value associated with a series of layer objects in the **Layers** collection in Corel PHOTO-PAINT, with the exception of the layers identified by their index numbers in the method parameters.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

You must reference the layers in the range by passing an index array as a parameter:

Parameters	Description
IndexArray	An **IndexArray** is a range of index numbers that identify and group specific layers in the **Layers** collection. By selecting a range of index numbers, you are selecting specific layer objects that become members of the array.

{button ,AL(`CLS_Layers;FNC_AllExcluding')} **Related Topics**

Layers.Merge

Sub **Merge**()

Description

The **Merge** method merges a layer in the **Layers** collection with the background in Corel PHOTO-PAINT. The manner of a merge depends on the merge mode of the active layer. A merge mode is an editing state that determines how the selected paint, object, or fill color combines with other colors in the image. For example, if you apply color or merge an object into the background using the Normal merge mode, the selected color replaces the original color in the image. If you apply color or merge an object into the background using the Add merge mode, the paint and paper colors are combined to produce a brighter color.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layers;FNC_Merge')} **Related Topics**

Layers.Paste

Function **Paste**([ByVal X As Variant], [ByVal Y As Variant], [ByVal IntoMask As Boolean = False]) As **LayerRange**

Description

The **Paste** method pastes an object into all layers in the Layers collection of Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
X	Specifies the horizontal location of the pasted object in the **Layers** collection. This value is optional and is measured in document units.
Y	Specifies the vertical location of the pasted object in the **Layers** collection. This value is optional and is measured in document units.
IntoMask	Specifies a True or False value that indicates in an object is pasted into a specific mask in **Layers** collection. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. This value is optional and the default value is False.

{button ,AL(`CLS_Layers;FNC_Paste')} **Related Topics**

Layers.Range

Function **Range**(ByRef **IndexArray**() As Variant) As **LayerRange**

Description

The **Range** method returns or sets a value associated with a series of layer objects in the **Layers** collection in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

The **Range** method allows you to reference several layers concurrently. The use of an index range makes it easier and faster to find, sort, and edit multiple layers in the collection.

You must reference the layers in the range by passing an index array as a parameter:

Parameters	Description
IndexArray	An **IndexArray** is a range of index numbers that identify and group specific layers in the **Layers** collection. By selecting a range of index numbers, you are selecting specific layer objects that become members of the array.

{button ,AL(`CLS_Layers;FNC_Range')} **Related Topics**

Layers.Selection

Function **Selection**() As **LayerRange**

Description

The **Selection** method creates a range of layers from the active selection of layers in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_Layers;FNC_Selection')} **Related Topics**

Mask properties

Mask Legend

- Application
- Floater
- Floating
- IsEmpty
  - Overlay
  - PaintOnMask
- Parent
  - PositionX
  - PositionY
  - SizeHeight
  - SizeWidth

Mask methods

Mask Legend

- AffineDistort
- AlignToDocument
- AlignToDocumentCenter
- AlignToGrid
- AlignToLayer
- AlignToLayerRange
- AlignToPoint
- Average
- Border
- CreateFromColorMask
- Delete
- Distort
- Expand
- Feather
- Flip
- GetPosition
- GetSize
- Grow
- Invert
- Load
- Move
- Reduce
- RemoveHoles
- Rotate
- SaveToNewChannel
- SelectAll
- SetPosition
- SetSize
- Similar
- Skew
- Smooth
- Stretch
- Threshold

Mask

Class **Mask**

Properties Methods Referenced By

The **Mask** class defines the characteristics of mask objects and describes the look and behavior of the objects through its properties and methods.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

A mask mode is a mask tool operation mode you must choose before you create or fine-tune a mask or selection. There are four mask modes: Normal, Additive, Subtractive, and XOR. The Normal mode (default) lets you select an area in an image. The Additive mode lets you expand the editable regions by selecting multiple areas in an image. The Subtractive mode lets you reduce the editable regions by removing areas from a selection. The XOR mode lets you select multiple areas in an image. If areas overlap, the overlapping regions are excluded from the selection and added to the mask.

{button ,AL(`CLS_Mask')} **Related Topics**

Mask.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Mask;FNC_Application')} **Related Topics**

Mask.Floater

Property **Floater** As **Floater**

Description

The **Floater** property returns a value associated with a floater selection in a mask in Corel PHOTO-PAINT. A floating selection is a selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

This property returns a Read-Only value.

{button ,AL(`CLS_Mask;FNC_Floater')} **Related Topics**

Mask.Floating

Property **Floating** As Boolean

Description

The **Floating** property returns a True or False value that indicates whether or not a mask is a floating selection in Corel PHOTO-PAINT. A floating selection is a selection that hovers or floats above an image and can be moved and modified without affecting the underlying pixels. You can paste information stored in the Clipboard as a floating selection in an image. When you defloat a selection, the pixels contained in the floating selection are merged with those in the underlying image.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

This property returns a Read-Only value.

{button ,AL(`CLS_Mask;FNC_Floating')} **Related Topics**

Mask.IsEmpty

Property **IsEmpty** As Boolean

Description

The **IsEmpty** property returns a True or False value that indicates whether or not a mask is empty in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

This property returns a Read-Only value.

{button ,AL(`CLS_Mask;FNC_IsEmpty')} **Related Topics**

Mask.Overlay

Property **Overlay** As Boolean

Description

The **IsEmpty** property returns a True or False value that indicates whether or not a mask is an overlay mask in Corel PHOTO-PAINT. An overlay mask is a red-tinted, transparent sheet that you can superimpose on the protected areas in an image. The mask overlay makes it easy to distinguish between the selected and the masked regions in an image. When the overlay is applied, the masked areas are displayed in varying degrees of red (according to their transparency). The deeper the saturation of the red tint, the greater the degree of protection.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

This property returns a Read-Only value.

{button ,AL(`CLS_Mask;FNC_Overlay')} **Related Topics**

Mask.PaintOnMask

Property **PaintOnMask** As Boolean

Description

The **PaintOnMask** property returns or sets a True or False value that indicates whether a mask is in Paint On Mask mode. The Paint On Mask mode lets you expand or reduce a selection by using a grayscale representation of the pixels: protected areas appear in black, editable areas in white, and areas that are partially masked or partially selected appear in gray.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_PaintOnMask')} **Related Topics**

Mask.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Mask;FNC_Parent')} **Related Topics**

Mask.PositionX

Property **PositionX** As Long

Description

The **PositionX** property returns or sets the horizontal position of a mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_PositionX')} **Related Topics**

Mask.PositionY

Property **PositionY** As Long

Description

The **PositionY** property returns or sets the vertical position of a mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_PositionY')} **Related Topics**

Mask.SizeHeight

Property **SizeHeight** As Long

Description

The **SizeHeight** property returns or sets the height of a mask object in Corel PHOTO-PAINT, measured in documents units. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_SizeHeight')} **Related Topics**

Mask.SizeWidth

Property **SizeWidth** As Long

Description

The **SizeWidth** property returns or sets the width of a mask object in Corel PHOTO-PAINT, measured in documents units. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_SizeWidth')} **Related Topics**

Mask.AffineDistort

Sub **AffineDistort**(ByVal **OffsetX** As Long, ByVal **OffsetY** As Long, ByVal **d11** As Double, ByVal **d12** As Double, ByVal **d21** As Double, ByVal **d22** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **AffineDistort** method distorts a mask in Corel PHOTO-PAINT. Distorting an object lets you stretch it non-proportionately. You can distort an object in the Image Window using the distortion handles. This method is used by CorelSCRIPT to record manual manipulation of a mask area. If you want to distort a mask in Corel SCRIPT, use the **MaskDistort** command.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
OffsetX	Sets the horizontal offset of the distortion.
OffsetY	Sets the vertical offset of the distortion.
d11	Sets a value that creates the distortion of one side of the layer.
d12	Sets a value that creates the distortion of one side of the layer.
d21	Sets a value that creates the distortion of one side of the layer.
d22	Sets a value that creates the distortion of one side of the layer.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Mask;FNC_AffineDistort')} **Related Topics**

Mask.AlignToDocument

Sub **AlignToDocument**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToDocument** method sets the alignment of a mask, according to the alignment type in a document in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_Mask;FNC_AlignToDocument')} **Related Topics**

Mask.AlignToDocumentCenter

Sub **AlignToDocumentCenter**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToDocumentCenter** aligns a mask to the center of a document in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_Mask;FNC_AlignToDocumentCenter')} **Related Topics**

Mask.AlignToGrid

Sub **AlignToGrid**(ByVal **Type** As **pntAlignType**)

Description

The **AlignToGrid** method aligns a mask to the grid in Corel PHOTO-PAINT. A Grid is a series of evenly spaced horizontal and vertical dots that are used to help draw and arrange objects. You can use the controls on the Grid and Ruler Setup dialog box to set the grid's parameters. For greater accuracy, you can also have objects in your illustration snap to the grid when you move, or draw them.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.

{button ,AL(`CLS_Mask;FNC_AlignToGrid')} **Related Topics**

Mask.AlignToLayer

Sub **AlignToLayer**(ByVal **Type** As **pntAlignType**, ByVal **Layer** As **Layer**)

Description

The **AlignToLayer** method aligns a mask to a layer in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
Layer	The **Layer** parameter identifies the layer to which the mask is aligned with the **AlignToLayer** method.

{button ,AL(`CLS_Mask;FNC_AlignToLayer')} **Related Topics**

Mask.AlignToLayerRange

Sub **AlignToLayerRange**(ByVal **Type** As **pntAlignType**, ByVal **LayerRange** As **LayerRange**)

Description

The **AlignToLayerRange** method aligns a mask to a range of layers in Corel PHOTO-PAINT. A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Type	The **Type** parameter sets the alignment type and returns a value of pntAlignType.
LayerRange	The LayerRange parameter identifies the range of layers to which the active layer is aligned with the AlignToLayerRange method.

{button ,AL(`CLS_Mask;FNC_AlignToLayerRange')} **Related Topics**

Mask.AlignToPoint

Sub **AlignToPoint**(ByVal **Type** As **pntAlignType**, ByVal **X** As Long, ByVal **Y** As Long)

Description

The **AlignToPoint** method aligns a mask to a specified point in Corel PHOTO-PAINT.A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Type	The Type parameter sets the alignment type and returns a value of pntAlignType.
X	Sets the horizontal coordinate of a point, measured in document units.
Y	Sets the horizontal coordinate of a point, measured in document units.

{button ,AL(`CLS_Mask;FNC_AlignToPoint')} **Related Topics**

Mask.Average

Sub **Average**(ByVal **Radius** As Long)

Description

The **Average** method samples all the pixels in the specified area of a mask and assigns an average color value to each one in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Radius	Defines the radius value in a mask.

{button ,AL(`CLS_Mask;FNC_Average')} **Related Topics**

pntBorderHard	0
pntBorderMedium	1
pntBorderSoft	2

Mask.Border

Sub **Border**(ByVal **Width** As Long, [ByVal Type As **pntBorderType** = pntBorderHard (0)])

Description

The **Border** method creates a new mask border along the edges of the original mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Width	Sets the width of the border.
Type	Sets the border type. This value returns pntBorderType. This value is optional and the default value is **pntBorderHard** (0).

{button ,AL(`CLS_Mask;FNC_Border')} **Related Topics**

Mask.CreateFromColorMask

Sub **CreateFromColorMask**(ByVal **ColorMask** As **ColorMask**, [ByVal MaskMode As **pntMaskMode** = pntMaskNormal (0)])

Description

The **CreateFromColorMask** method creates a new mask from a color mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A color mask is a mask that you use to protect colors in an image. Only pixels that fall within the color range you specify are included in the color mask. You can use the Magic Wand Mask tool, the Lasso Mask tool, or the Color Mask command to create color masks.

Parameters	Description
ColorMask	Specifies the color mask to use when creating a new mask.
MaskMode	Sets the mask mode when creating a new mask. This value returns pntMaskMode. This value is optional and the default value is **pntMaskNormal** (0).

{button ,AL(`CLS_Mask;FNC_CreateFromColorMask')} **Related Topics**

Mask.Delete

Sub **Delete**()

Description

The **Delete** method deletes a mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_Delete')} **Related Topics**

Mask.Distort

Sub **Distort**(ByVal **x1** As Long, ByVal **y1** As Long, ByVal **x2** As Long, ByVal **y2** As Long, ByVal **x3** As Long, ByVal **y3** As Long, ByVal **x4** As Long, ByVal **y4** As Long, [ByVal AntiAlias As Boolean = True])

Description

The **Distort** method distorts the shape of the current mask in Corel PHOTO-PAINT. Distorting an object lets you stretch it non-proportionately. You can distort an object in the Image Window using the distortion handles.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
x1	Sets the distortion value for the horizontal coordinate of a mask's corner.
y1	Sets the distortion value for the vertical coordinate of a mask's corner.
x2	Sets the distortion value for the horizontal coordinate of a mask's corner.
y2	Sets the distortion value for the vertical coordinate of a mask's corner.
x3	Sets the distortion value for the horizontal coordinate of a mask's corner.
y3	Sets the distortion value for the vertical coordinate of a mask's corner.
x4	Sets the distortion value for the horizontal coordinate of a mask's corner.
y4	Sets the distortion value for the vertical coordinate of a mask's corner.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Mask;FNC_Distort')} **Related Topics**

Mask.Expand

Sub **Expand**(ByVal **Width** As Long)

Description

The **Expand** method extends transparent areas of the mask into opaque areas by a given number of pixels in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Width	Defines the new with of the mask, measured in pixels.

{button ,AL(`CLS_Mask;FNC_Expand')} **Related Topics**

Mask.Feather

Sub **Feather**(ByVal **Width** As Long, [ByVal Type As **pntFeatherType** = pntFeatherAverage (0)], [ByVal Edges As **pntEdgeType** = pntEdgeLinear (0)])

Description

The **Feather** method feathers the edges of a mask in Corel PHOTO-PAINT. Feathering is the gradual blending of pixels between a selection or an object and the surrounding background. Feathering produces a softer, more natural-looking edge.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Width	Sets the width of a feather effect of a mask's edges.
Type	Sets the feather type, which returns a value of pntFeatherType. This value is optional and the default value is pntFeatherAverage (0).
Edges	Sets the feather edge, which returns a value of pntEdgeType. This value is optional and the default value is pntEdgeLinear (0)

{button ,AL(`CLS_Mask;FNC_Feather')} **Related Topics**

Mask.Flip

Sub **Flip**(ByVal **Axes** As **cdrFlipAxes**)

Description

The **Flip** method flips a mask object in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

A mask can be flipped horizontally, vertically, or both.

Parameters	Description
Axes	Sets the manner of the flip in the layer object. This value returns cdrFlipAxes.

{button ,AL(`CLS_Mask;FNC_Flip')} **Related Topics**

Mask.GetPosition

Sub **GetPosition**(ByRef **PositionX** As Long, ByRef **PositionY** As Long)

Description

The **GetPosition** method gets the position of a mask object in Corel PHOTO-PAINT.A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
PositionX	Gets the horizontal position of the mask's center point, in document units.
PositionY	Gets the vertical position of the mask's center point, in document units. .

{button ,AL(`CLS_Mask;FNC_GetPosition')} **Related Topics**

Mask.GetSize

Sub **GetSize**(ByRef **Width** As Long, ByRef **Height** As Long)

Description

The **GetSize** method gets the width and height of a mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Width	The **Width** parameter gets the width of a mask, measured in document units.
Height	The **Height** parameter gets the height of a mask, measured in document units.

{button ,AL(`CLS_Mask;FNC_GetSize')} **Related Topics**

Mask.Grow

Sub **Grow**([ByVal AntiAlias As Boolean = True], [ByVal UseLayer As Boolean = False], [ByVal Layer As **Layer** = 0], [ByVal Tolerance As **pntToleranceMode** = pntToleranceNormal (0)], [ByVal Normal As Long = 0], [ByVal Hue As Long = 0], [ByVal Saturation As Long = 0], [ByVal Brightness As Long = 0])

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.

Member of **Mask**

Parameters	Description
AntiAlias	Description of AntiAlias goes here (**in**) Optional Default value = **True**
UseLayer	Description of UseLayer goes here (**in**) Optional Default value = **False**
Layer	Description of Layer goes here (**in**) Optional Default value = **0**
Tolerance	Description of Tolerance goes here (**in**) Optional Default value = **pntToleranceNormal** (0)
Normal	Description of Normal goes here (**in**) Optional Default value = **0**
Hue	Description of Hue goes here (**in**) Optional Default value = **0**
Saturation	Description of Saturation goes here (**in**) Optional Default value = **0**
Brightness	Description of Brightness goes here (**in**) Optional Default value = **0**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Mask;FNC_Grow')} **Related Topics**

Mask.Invert

Sub **Invert**()

Description

The **Invert** method reverses the area included in the mask in Corel PHOTO-PAINT. Often, it is easier to select the areas you want to protect from changes and then invert the selection. For example, if you want to edit a complex, irregularly shaped area, you can begin by selecting the surrounding area. Then, you can reverse the mask so that the area that was initially selected is protected, and the area that was masked is selected, i.e., available for editing.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_Invert')} **Related Topics**

Mask.Load

Sub **Load**(ByVal **FileName** As String, [ByVal LoadMode As **pntLoadMode** = pntLoadAll (0)], [ByVal Left As Long = 0], [ByVal Top As Long = 0], [ByVal Width As Long = 0], [ByVal Height As Long = 0], [ByVal CropLeft As Long = 0], [ByVal CropTop As Long = 0], [ByVal CropWidth As Long = 0], [ByVal CropHeight As Long = 0])

Description

The **Load** method loads a mask from a file outside of Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
FileName	Specifies the computer location of the file containing the mask object that is loaded into Corel PHOTO-PAINT.
LoadMode	Sets the load mode of the mask. This value returns pntLoadMode. This value is optional and the default value is pntLoadAll (0).
Left	Sets the left position of the mask. This value is optional and the default value is 0.
Top	Sets the top position of the mask. This value is optional and the default value is 0.
Width	Sets the width of the mask. This value is optional and the default value is 0.
Height	Sets the height of the mask. This value is optional and the default value is 0.
CropLeft	Sets the left position of the mask if cropped. This value is optional and the default value is 0.
CropTop	Sets the top position of the mask if cropped. This value is optional and the default value is 0.
CropWidth	Sets the width of the mask if cropped. This value is optional and the default value is 0.
CropHeight	Sets the height of the mask if cropped. This value is optional and the default value is 0.

{button ,AL(`CLS_Mask;FNC_Load')} **Related Topics**

Mask.Move

Sub **Move**(ByVal **Dx** As Long, ByVal **Dy** As Long)

Description

The **Move** method changes the position of a mask object by moving it a specified distance from its original location in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Dx	The **Dx** parameter specifies the horizontal distance to move a mask with the **Move** method. Delta refers to a limited incremental value in a variable.
Dy	The **Dy** parameter specifies the vertical distance to move a mask with the **Move** method. Delta refers to a limited incremental value in a variable.

{button ,AL(`CLS_Mask;FNC_Move')} **Related Topics**

Mask.Reduce

Sub **Reduce**(ByVal **Width** As Long)

Description

The **Reduce** method extends opaque areas of the mask into transparent areas by a given number of pixels in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Width	Sets the new width of the mask, measured in pixels.

{button ,AL(`CLS_Mask;FNC_Reduce')} **Related Topics**

Mask.RemoveHoles

Sub **RemoveHoles**()

Description

The **RemoveHoles** method removes the holes in a mask in Corel PHOTO-PAINT. When you use the Lasso Mask tool, the Magic Wand Mask tool, or the Color Mask command to select areas on an image, you often end up with masked regions that are completely surrounded by the selection. You can unmask these "holes" and make them part of the selection and editable.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_RemoveHoles')} **Related Topics**

Mask.Rotate

Sub **Rotate**(ByVal **Angle** As Double, ByVal **CenterX** As Long, ByVal **CenterY** As Long, [ByVal AntiAlias As Boolean = True])

Description

The **Rotate** method rotates the mask a specified degree in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Angle	Sets the angle of rotation, in degrees.
CenterX	Sets the horizontal position for the center of rotation.
CenterY	Sets the vertical position for the center of rotation.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Mask;FNC_Rotate')} **Related Topics**

Mask.SaveToNewChannel

Function **SaveToNewChannel**([ByVal Name As String], [ByVal Opacity As Long = 50], [ByVal Color As **Color** = 0], [ByVal InvertOverlay As Boolean = False]) As **Channel**

SAVE TO NEW CHANNEL

Member of **Mask**

Parameters	Description
Name	Description of Name goes here (**in**) Optional
Opacity	Description of Opacity goes here (**in**) Optional Default value = **50**
Color	Description of Color goes here (**in**) Optional Default value = **0**
InvertOverlay	Description of InvertOverlay goes here (**in**) Optional Default value = **False**

Return value

Returns **Channel**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Mask;FNC_SaveToNewChannel')} **Related Topics**

Mask.SelectAll

Sub **SelectAll**()

Description

The **SelectAll** method selects all mask objects in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

{button ,AL(`CLS_Mask;FNC_SelectAll')} **Related Topics**

Mask.SetPosition

Sub **SetPosition**(ByVal **PositionX** As Long, ByVal **PositionY** As Long)

Description

The **SetPosition** method sets the position of a mask object in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
PositionX	Sets the horizontal position of the mask's center point, in document units.
PositionY	Sets the vertical position of the mask's center point, in document units. .

{button ,AL(`CLS_Mask;FNC_SetPosition')} **Related Topics**

Mask.SetSize

Sub **SetSize**([ByVal Width As Long = 0], [ByVal Height As Long = 0])

Description

The **SetSize** method sets the width and height of a mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Width	The **Width** parameter sets the width of a mask, measured in document units.
Height	The **Height** parameter sets the height of a mask, measured in document units.

{button ,AL(`CLS_Mask;FNC_SetSize')} **Related Topics**

Mask.Similar

Sub **Similar**([ByVal AntiAlias As Boolean = True], [ByVal UseLayer As Boolean = False], [ByVal Layer As **Layer** = 0], [ByVal Tolerance As **pntToleranceMode** = pntToleranceNormal (0)], [ByVal Normal As Long = 0], [ByVal Hue As Long = 0], [ByVal Saturation As Long = 0], [ByVal Brightness As Long = 0])

Adds to the current mask area any areas of the image having colors similar to the image color within the current mask area.

Member of **Mask**

Parameters	Description
AntiAlias	Description of AntiAlias goes here (**in**) Optional Default value = **True**
UseLayer	Description of UseLayer goes here (**in**) Optional Default value = **False**
Layer	Description of Layer goes here (**in**) Optional Default value = **0**
Tolerance	Description of Tolerance goes here (**in**) Optional Default value = **pntToleranceNormal** (0)
Normal	Description of Normal goes here (**in**) Optional Default value = **0**
Hue	Description of Hue goes here (**in**) Optional Default value = **0**
Saturation	Description of Saturation goes here (**in**) Optional Default value = **0**
Brightness	Description of Brightness goes here (**in**) Optional Default value = **0**

Example

Example of usage goes here

```
Code line
```

{button ,AL(`CLS_Mask;FNC_Similar')} **Related Topics**

Mask.Skew

Sub **Skew**(ByVal **HAngle** As Double, ByVal **VAngle** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **Skew** method skews the active mask in Corel PHOTO-PAINT. Skewing refers to a slant of an object. Skewing a layer slants the layer in Corel PHOTO-PAINT.

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
HAngle	The **HAngle** parameter specifies the degree in which to slant the mask horizontally.
VAngle	The **VAngle** parameter specifies the degree in which to slant the mask vertically.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Mask;FNC_Skew')} **Related Topics**

Mask.Smooth

Sub **Smooth**(ByVal **Radius** As Long)

Description

The **Smooth** method smoothes over or rounds off the sharp angles of a mask in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Radius	Sets the degree of smoothness in the mask.

{button ,AL(`CLS_Mask;FNC_Smooth')} **Related Topics**

Mask.Stretch

Sub **Stretch**(ByVal **XOrigin** As Long, ByVal **YOrigin** As Long, ByVal **XScale** As Double, ByVal **YScale** As Double, [ByVal AntiAlias As Boolean = True])

Description

The **Stretch** method stretches the active mask in Corel PHOTO-PAINT. Stretching an object means to size an object horizontally or vertically. Stretching changes the size of an object in one direction only, as opposed to sizing, which maintains the aspect ratio (the ratio of height to width).

A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
XOrigin	The **XOrigin** parameter specifies the horizontal position of the anchor point when stretching a mask. The anchor point is the point that remains stationary when you stretch, scale, mirror, or skew an object. This value is optional and the default value is 0. This value is measured in document units.
YOrigin	The **YOrigin** parameter specifies the vertical position of the anchor point when stretching a mask. The anchor point is the point that remains stationary when you stretch, scale, mirror, or skew an object. This value is optional and the default value is 0. This value is measured in document units.
XScale	The **XScale** parameter specifies the width value of the aspect ratio used when stretching objects. The aspect ratio is the ratio of the width of an image to its height (expressed mathematically as x:y). For example, the aspect ratio of an image that is 640 x 480 pixels is 4:3. This value is optional and the default value is 100.
YScale	The **YScale** parameter specifies the width value of the aspect ratio used when stretching objects. The aspect ratio is the ratio of the width of an image to its height (expressed mathematically as x:y). For example, the aspect ratio of an image that is 640 x 480 pixels is 4:3. This value is optional and the default value is 100.
AntiAlias	The **AntiAlias** parameter sets a True or False value that indicates whether or not selections with curved and diagonal lines have their jagged edges removed. Anti-Aliasing is a method of smoothing the curved and diagonal edges contained in bitmap images. Anti-aliasing partially fills intermediate pixels along those edges to smooth the transition between the edge and the surrounding image. Anti-aliasing reduces or eliminates jagged edges. This value is optional and the default value is True.

{button ,AL(`CLS_Mask;FNC_Stretch')} **Related Topics**

Mask.Threshold

Sub **Threshold**(ByVal **Threshold** As Long)

Description

The **Threshold** method converts grayscale to a black and white (or bi-level) mask using the specified threshold value in Corel PHOTO-PAINT. A mask is a selection tool that isolates the area that you want to protect from changes when you apply color, filters, or other effects to an image. A mask acts as a protective layer or sheet that covers the area on an image that should not be affected by your editing. When you select an area on an image using a mask tool, this area is available for editing, while the rest of the image is masked, i.e., protected from changes. You can create regular and color masks.

Parameters	Description
Threshold	The **Threshold** parameter lets you change threshold values for the colors in an image by converting a range of colors to black or white. The threshold values that you set determine which pixels become black and which pixels become white. If you convert colors to black, only pixels with a brightness value that is lower than the threshold value are converted to black. If you convert colors to white, only pixels with a brightness value that is higher than the threshold value are converted to white. The Threshold brightness values range from 0 (black) to 255 (white).

{button ,AL(`CLS_Mask;FNC_Threshold')} **Related Topics**

ObjectLayer properties

ObjectLayer Legend

- Application
- ClipMask
- DropShadow
- IsTextLayer
- Parent

ObjectLayer methods

ObjectLayer Legend

- AddClipMask
- Defringe
- RemoveMatte

ObjectLayer

Class **ObjectLayer**

Properties Methods Referenced By

The **ObjectLayer** class defines the characteristics of object layers and describes the look and behavior of the objects through its properties and methods. An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_ObjectLayer')} **Related Topics**

ObjectLayer.Application

Property **Application** As **Application**

The **ObjectLayer** class defines the characteristics of object layers and describes the look and behavior of the objects through its properties and methods. An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprinttable. Use layers to help you organize different components of a complex image.

{button ,AL(`CLS_ObjectLayer;FNC_Application')} **Related Topics**

ObjectLayer.ClipMask

Property **ClipMask** As **ClipMask**

Description

The **ClipMask** property returns a value associated with a clip mask object on the object layer in Corel PHOTO-PAINT. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_ObjectLayer;FNC_ClipMask')} **Related Topics**

ObjectLayer.DropShadow

Property **DropShadow** As **DropShadow**

Description

The **DropShadow** property returns a value associated with a drop shadow effect object on the object layer in Corel PHOTO-PAINT. By adding drop shadows, you create the illusion of depth between objects. You can create two types of object drop shadows: flat and perspective. Flat drop shadows silhouette objects. Perspective drop shadows create three-dimensional depth.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_ObjectLayer;FNC_DropShadow')} **Related Topics**

ObjectLayer.IsTextLayer

Property **IsTextLayer** As Boolean

Description

The **IsTextLayer** property returns a True or False value that indicates whether or not an object layer is a text layer in Corel PHOTO-PAINT. An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

This property returns a Read-Only value.

{button ,AL(`CLS_ObjectLayer;FNC_IsTextLayer')} **Related Topics**

ObjectLayer.Parent

Property **Parent** As **Layer**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_ObjectLayer;FNC_Parent')} **Related Topics**

pntFromMask	0
pntFromInvertedMask	1
pntFromObjectTransparency	2
pntToShowAll	3
pntToHideAll	4
pntUseOpacity	5

ObjectLayer.AddClipMask

Sub **AddClipMask**(ByVal **CreationMode** As **pntClipMaskMode**, [ByVal OpacityLevel As Long = 0])

Description

The **AddClipMask** method adds a clip masks object to an object layer in Corel PHOTO-PAINT. A clip mask is a mask that lets you edit an object's transparency levels without affecting the pixels in the object. You can change the transparency levels directly on the object and then add the clip mask, or add the clip mask before making the changes.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
CreationMode	Sets the creation mode for the new clip mask on the object layer. This value returns pntClipMaskMode.
OpacityLevel	Sets the level of opacity in the new clip mask on the object layer. This value is optional and the default value is 0. Opacity is the opposite of transparency. If an area is 100% opaque, you cannot see through it. Levels under 100% increase the ability to see through objects.

{button ,AL(`CLS_ObjectLayer;FNC_AddClipMask')} **Related Topics**

ObjectLayer.Defringe

Sub **Defringe**(ByVal **Amount** As Long)

Description

The **Defringe** method defringes the edges of the object layer in Corel PHOTO-PAINT. The **Defringe** method replaces the color of the pixels that lie along the outside edge of an object with the color of the pixels that lie inside the boundaries of the object.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

An object can have stray pixels along its edges; these strays contrast with neighboring pixels, giving the object a slightly ragged effect. Defringing stray pixels blends the object with the background to create a smooth, polished appearance.

Parameters	Description
Amount	Sets the level of the defringe effect in the edges of the object layer. Large defringe values create a gradual transition between the edges of the object and the background. Values range from 1 to 100.

{button ,AL(`CLS_ObjectLayer;FNC_Defringe')} **Related Topics**

ObjectLayer.RemoveMatte

Sub **RemoveMatte**(ByVal **Type** As **pntMatteType**)

Description

The **RemoveMatte** method removes a matte from the object layer in Corel PHOTO-PAINT. You can change the transparency of the pixels along the edge of a feathered object using the Remove Matte commands. The more transparent the pixels are, the more you can see through them. The Remove Black Matte command makes semitransparent pixels more transparent. The Remove White Matte command makes semitransparent pixels less transparent.

An object is an independent bitmap that is layered above the background image. Transformations applied to objects do not affect the underlying image. The object layer is the layer containing an object in Corel PHOTO-PAINT.

A layer is one of a series of transparent planes on which you can place objects in an image. You can control how objects in your image overlay one another by moving the layer and the objects they contain. You can also lock layers and make them invisible and nonprintable. Use layers to help you organize different components of a complex image.

Parameters	Description
Type	Sets the type of matte that is removed from an object layer. This value returns pntMatteType.

{button ,AL(`CLS_ObjectLayer;FNC_RemoveMatte')} **Related Topics**

pntBlackMatte	0
pntWhiteMatte	1

Palette properties

Palette Legend

- Application
  Color
- ColorCount
- DuplicatePresent
  Name
- PaletteID
- Parent
- Type

Palette methods

Palette Legend

- AddColor
- Close
- Colors
- GetIndexOfColor
- InsertColor
- RemoveColor
- Save

Palette

Class **Palette**

Properties Methods Referenced By

The **Palette** class defines the characteristics of palette objects and describes the look and behavior of the objects through its properties and methods.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The on-screen color palette is a toolbar that displays a series of color swatches. It is used to select colors for use in Corel PHOTO-PAINT. You can display multiple on-screen Color Palettes. They can be docked or left floating in the Application Window.

The Default on-screen Color Palette is the color palette that you will see in any of the drop down color pickers. You can select any Color Palette to be the Default on-screen Color Palette.

You have the ability to display multiple on-screen Color Palettes, and keep them floating, or dock them to any edge of the Application Window.

There are two types of color palettes from which you can choose colors: fixed color palettes and custom color palettes. On-screen Color Palettes are used to display and select colors from both fixed and custom color palettes.

Fixed color palettes

Fixed color palettes are provided by third-party manufacturers and are most useful when accompanied by a color swatch book. A swatch book is a collection of color samples that shows exactly what each color looks like when printed.

Custom color palettes

Custom color palettes are collections of colors you have chosen to save as a color palette file (.CPL extension).You have the ability to copy a color swatch by dragging a color swatch from any palette into your custom palettes. There is no limit to the number of custom palettes you can create.

When you create a custom palette, the palette is empty and ready for you to choose the colors you want to include in it. You can create a custom palette that includes all the colors from the object that you selected or from the current document.

{button ,AL(`CLS_Palette')} **Related Topics**

Palette.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub PaletteApp()
With ActivePalette
    MsgBox .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_Application')} **Related Topics**

Palette.Color

Property **Color**(ByVal **Index** As Long) As **Color**

Description

The **Color** property returns or set a value associated with a specific color in a palette in Corel PHOTO-PAINT. The color is identified by its index number.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

You must reference a color in the palette by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each color object in a Palette; it uniquely identifies each color in the palette object.

Example

The following code example displays the name of the first color in the active palette in a message box:

```
Sub PaletteColorName()
With ActivePalette
    MsgBox .Color(1).Name
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_Color')} **Related Topics**

Palette.ColorCount

Property **ColorCount** As Long

Description

The **ColorCount** property returns a value associated with the number of colors in a color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The **ColorCount** property returns a Read-Only value.

Example

The following code example displays the total number of colors found in the active palette of Corel PHOTO-PAINT:

```
Sub PaletteColorCount()
With ActivePalette
    MsgBox .ColorCount
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_ColorCount')} **Related Topics**

Palette.DuplicatePresent

Property **DuplicatePresent** As Boolean

Description

The **DuplicatePresent** property returns a True or False value indicating if a color is repeated in a color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

If the value returned is True, the color is duplicated in the color palette. If the value is False, the color is unique within the palette.

The **DuplicatePresent** property returns a Read-Only value.

Example

The following code example displays a true or false value in a message box, indicating the presence of a duplicated color in the active palette of Corel PHOTO-PAINT:

```
Sub PaletteDuplicate()
With ActivePalette
    MsgBox .DuplicatePresent
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_DuplicatePresent')} **Related Topics**

Palette.Name

Property **Name** As String

Description

The **Name** property returns or sets a string value associated with the name of a palette object in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The palette name can only be set with the **Name** property if the palette is a custom palette.

Example

The following code example displays the name of the active palette in Corel PHOTO-PAINT:

```
Sub PaletteName()
With ActivePalette
    MsgBox .Name
End With
End Sub
Code line
```

{button ,AL(`CLS_Palette;FNC_Name')} **Related Topics**

Palette.PaletteID

Property **PaletteID** As **cdrPaletteID**

Description

The **PaletteID** property returns a value associated with the type of a palette in Corel PHOTO-PAINT. The value returned is cdrPaletteID.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The **PaletteID** returns a Read-Only value.

Example

The following code example displays the ID number of the active palette in a message box. The ID number references a type of palette in Corel PHOTO-PAINT:

```
Sub IDPalette()
With ActivePalette
    MsgBox .PaletteID
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_PaletteID')} **Related Topics**

Palette.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the name of the parent object of a color in the active palette of Corel PHOTO-PAINT:

```
Sub PaletteParent(0
With ActivePalette
    MsgBox .Colors.Parent.Name
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_Parent')} **Related Topics**

Palette.Type

Property **Type** As **cdrPaletteType**

Description

The **Type** property returns a value indicating if a color palette is a custom or fixed color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The value returned is cdrPaletteType.

The **Type** property returns a Read-Only value.

Example

The following code example displays the type of the active palette in Corel PHOTO-PAINT:

```
Sub PaletteType()
With ActivePalette
    MsgBox .Type
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_Type')} **Related Topics**

Palette.AddColor

Sub **AddColor**(ByVal **Color** As **Color**)

Description

The **AddColor** method adds a color swatch to a custom palette in Corel PHOTO-PAINT.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

In order to add a color to a custom palette, you must pass the color name as a parameter in the **AddColor** method:

Parameters	Description
Color	Color creates a new color in a palette object in Corel PHOTO-PAINT.

Example

The following code example adds a new color to the active color palette using the **CreateColorEx** method. The new color is added to the active color palette in Corel PHOTO-PAINT:

```
Sub CreatePaletteColor()
ActivePalette.AddColor CreateColorEx(2, 90, 90, 0, 0)
End Sub
```

{button ,AL(`CLS_Palette;FNC_AddColor')} **Related Topics**

Palette.Close

Sub **Close**()

Description

The **Close** method closes a specified color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Example

The following code example closes the active palette in Corel PHOTO-PAINT:

```
Sub PaletteClose()
ActivePalette.Close
End Sub
```

{button ,AL(`CLS_Palette;FNC_Close')} **Related Topics**

Palette.Colors

Function **Colors**() As **Colors**

Description

The **Colors** property returns or sets a value associated with the Colors collection in a palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Example

The following code example displays the number of colors in the colors collection of the active palette in Corel PHOTO-PAINT:

```
Sub PaletteColors()
With ActivePalette
    MsgBox .Colors.Count
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_Colors')} **Related Topics**

Palette.GetIndexOfColor

Function **GetIndexOfColor**(ByVal **Color** As **Color**) As Long

Description

The **GetIndexOfColor** property returns a value associated with the index number of a color in a palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The **GetIndexOfColor** property returns a Read-Only value.

Parameters	Description
Color	Specifies a color swatch in a palette object in Corel PHOTO-PAINT.

{button ,AL(`CLS_Palette;FNC_GetIndexOfColor')} **Related Topics**

Palette.InsertColor

Sub **InsertColor**(ByVal **Index** As Long, ByVal **Color** As **Color**)

Description

The **InsertColor** method places an existing color into a custom palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

In order to insert a color with the **InsertColor** method, you must past the index and color values as parameters:

Parameters	Description
Index	**Index** is a preset placeholder for each color object in a Palette; it uniquely identifies each color in the palette object.
Color	**Color** creates a new color in a palette object in Corel PHOTO-PAINT.

{button ,AL(`CLS_Palette;FNC_InsertColor')} **Related Topics**

Palette.RemoveColor

Sub **RemoveColor**(ByVal **Index** As Long)

Description

The **RemoveColor** method removes a color from a custom palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

In order to remove a color with the **RemoveColor** method, you must pass the color's index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each color object in a Palette; it uniquely identifies each color in the palette object.

Example

The following code example removed the first color from the active palette in Corel PHOTO-PAINT:

```
Sub PaletteRemove()
With ActivePalette
    .RemoveColor(1)
End With
End Sub
```

{button ,AL(`CLS_Palette;FNC_RemoveColor')} **Related Topics**

Palette.Save

Sub **Save**()

Description

The **Save** method saves a color palette for future use in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

Example

The following code example saves the current color palette in Corel PHOTO-PAINT:

```
Sub PaletteSave()
ActivePalette.Save
End Sub
```

{button ,AL(`CLS_Palette;FNC_Save')} **Related Topics**

Fixed color palettes are provided by third-party manufacturers and are most useful when accompanied by a color swatch book. A swatch book is a collection of color samples that shows exactly what each color looks like when printed.

Custom color palettes are collections of colors you have chosen to save as a color palette file (.CPL extension).You have the ability to copy a color swatch by dragging a color swatch from any palette into your custom palettes. There is no limit to the number of custom palettes you can create.

When you create a custom palette, the palette is empty and ready for you to choose the colors you want to include in it. You can create a custom palette that includes all the colors from the object that you selected or from the current document.

cdrFixedPalette=0

cdrCustomPalette=1

Palettes properties

Palettes Legend

- Application
- Count
-
- Item
- Parent

Palettes methods

Palettes Legend

- Create
- Open
- OpenFixed

Palettes

Class **Palettes**

Properties Methods Referenced By

The **Palette** class defines the characteristics of **Palettes** collection objects and describes the look and behavior of the objects through its properties and methods.

A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The on-screen color palette is a toolbar that displays a series of color swatches. It is used to select colors for use in Corel PHOTO-PAINT. You can display multiple on-screen Color Palettes. They can be docked or left floating in the Application Window.

The Default on-screen Color Palette is the color palette that you will see in any of the drop down color pickers. You can select any Color Palette to be the Default on-screen Color Palette.

You have the ability to display multiple on-screen Color Palettes, and keep them floating, or dock them to any edge of the Application Window.

There are two types of color palettes from which you can choose colors: fixed color palettes and custom color palettes. On-screen Color Palettes are used to display and select colors from both fixed and custom color palettes.

Fixed color palettes

Fixed color palettes are provided by third-party manufacturers and are most useful when accompanied by a color swatch book. A swatch book is a collection of color samples that shows exactly what each color looks like when printed.

Custom color palettes

Custom color palettes are collections of colors you have chosen to save as a color palette file (.CPL extension).You have the ability to copy a color swatch by dragging a color swatch from any palette into your custom palettes. There is no limit to the number of custom palettes you can create.

When you create a custom palette, the palette is empty and ready for you to choose the colors you want to include in it. You can create a custom palette that includes all the colors from the object that you selected or from the current document.

{button ,AL(`CLS_Palettes')} **Related Topics**

Palettes.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub PalettesApp()
With Palettes
    MsgBox "You are using Corel PHOTO-PAINT " & .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Palettes;FNC_Application')} **Related Topics**

Palettes.Count

Property **Count** As Long

Description

The **Count** property returns the number of palettes in the **Palettes** collection of Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

The **Count** property returns a Read-Only value.

Example

The following code example displays the number of open color palettes in the current session of Corel PHOTO-PAINT in a message box:

```
Sub PalettesCount()
MsgBox Palettes.Count
End Sub
```

{button ,AL(`CLS_Palettes;FNC_Count')} **Related Topics**

Palettes.Item

Property **Item**(ByVal **IndexOrName** As Variant) As **Palette**

Description

The **Item** property returns a value associated with the index number of a Palette object in the **Palettes** collection of Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

`Palettes(2)` refers to the second palette object in the **Palettes** collection of Corel PHOTO-PAINT.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Palettes.Item(2)` is the same as `Palettes(2)` - they both reference the second palette object in the **Palettes** collection.

You must reference a palette in the collection by passing its index number or name as a parameter:

Parameters	Description
IndexOrName	**Index** is a preset placeholder for each object in a **Palettes** collection; it uniquely identifies each member of the collection. **Name** is the unique text name given to each palette.

Example

The following code example displays the number of colors in the first palette in the Palettes collection in a message box:

```
Sub PalettesItem()
With Palettes.Item(1)
'Palettes(1) may be used here
    MsgBox .Colors.Count
End With
End Sub
```

{button ,AL(`CLS_Palettes;FNC_Item')} **Related Topics**

Palettes.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the name of the parent object of the first palettes' colors in a message box:

```
Sub PalettesParent()
With Palettes.Item(1)
    MsgBox .Colors.Parent.Name
End With
End Sub
```

{button ,AL(`CLS_Palettes;FNC_Parent')} **Related Topics**

Palettes.Create

Function **Create**(ByVal **Name** As String, [ByVal FileName As String], [ByVal Overwrite As Boolean = False]) As **Palette**

Palettes

Description

The **Create** method creates a new empty custom color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper. Once a new palette is created, you can add colors to it according to your needs.

In order to create a new custom palette with the **Create** method, you must pass a palette name and an overwrite specification as parameters:

Parameters	Description
Name	**Name** is a string value that identifies the new custom palette. Since the palette is new, you must give it a new name when you use the **Create** method.
Overwrite	The **Overwrite** parameter allows you to remove the existing palette in Corel PHOTO-PAINT and replace it with the new palette created with the **Create** method. By setting the value to True, you will replace the default palette with the new palette. If set to False, the new palette co-exists with the default palette. The default value is False.

Example

The following code example creates an empty custom color palette in Corel PHOTO-PAINT. Since the Overwrite parameter is False by default, it does not need to be specified in the expression:

```
Sub PalettesCreate()
Palettes.Create ("My New Palette")
End Sub
```

{button ,AL(`CLS_Palettes;FNC_Create')} **Related Topics**

Palettes.Open

Function **Open**(ByVal **FileName** As String) As **Palette**

Description

The **Open** method opens an existing color palette in Corel PHOTO-PAINT. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

In Corel PHOTO-PAINT, you may have several palettes open at the same time. Opening a palette will not close the currently opened palette.

In order to open a color palette with the **Open** method, you must specify the file name of the palette as a parameter:

Parameters	Description
FileName	**FileName** refers to the full path name (with a .cpl extension) of a color palette in Corel PHOTO-PAINT. It is a unique identifier of each palette in the **Palettes** collection. If the Default CMYK Palette is stored on the root of C, its FileName is C:\Default CMYK Palette.cpl

Example

The following code example opens the Gray256 palette in Corel PHOTO-PAINT.

```
Sub PalettesOpen()
Palettes.Open ("C:\ProgramFiles\Corel\Graphics10\Custom\Palettes\Gray256.cpl")
End Sub
```

{button ,AL(`CLS_Palettes;FNC_Open')} **Related Topics**

Palettes.OpenFixed

Function **OpenFixed**(ByVal **PaletteID** As **cdrPaletteID**) As **Palette**

Description

The **OpenFixed** method opens an existing fixed palette in Corel PHOTO-PAINT, according to its palette ID.. A Color Palette is a collection of solid colors. In Corel PHOTO-PAINT, you can use the on-screen Color Palette, the Select Color dialog box, or the Color Docker to choose colors for fills, outlines, and paper.

In order to open a fixed palette with the **OpenFixed** method, you must pass the ID of the palette as a parameter.

When passing the **PaletteID** parameter, you have two options. You may pass the numerical ID of the fixed palette or you may pass the Corel PHOTO-PAINT constant name of the fixed palette. For example, `OpenFixed (7)` and `OpenFixed (cdrUniform)` both open the Uniform Colors fixed palette in Corel PHOTO-PAINT.

Parameters	Description
PaletteID	The **PaletteID** property returns a value associated with the type of a palette in Corel PHOTO-PAINT. The value returned is cdrPaletteID.

Example

The following code example opens the Uniform Colors fixed palette in Corel PHOTO-PAINT:

```
Sub OpenFixedPalette()
Palettes.OpenFixed (cdrUniform)
'you may also use Palettes.OpenFixed (7)
End Sub
```

{button ,AL(`CLS_Palettes;FNC_OpenFixed')} **Related Topics**

Tools properties

Tools Legend

- Application
- Count
-
- Item
- Parent

Tools

Class **Tools**

Properties Referenced By

The **Tools** class defines the characteristics of tool objects and describes the look and behavior of the collection objects through its properties and methods. The Corel PHOTO-PAINT Toolbox contains tools for creating and manipulating objects and selections in your image. The zoom tools let you view specific areas of your image, and the shaping and painting tools let you modify your image. The Toolbox also contains tools that let you apply modifications interactively.

The Standard toolbar, located at the top of the Application Window by default, gives you access to some of the most common application commands, such as opening and saving images. The Property Bar and Docker windows provide access to commands that are relevant to the active tool or current task and can be opened, closed, and moved across your screen.

The tool cursor is a small version of the tool icon that represents all tools in the Image Window. This lets you quickly see which tool is currently selected by looking at the cursor in the Image Window. Shape and Mask tools are displayed as a crosshair cursor with a small representation of the tool on the top-right section of the crosshair. The Text tool is always represented by an I-beam, the Object Picker tool by an arrow.

Each button on a toolbar represents a command. Some are shortcuts to menu commands; others are commands that are available only as toolbar buttons. You can customize your work area by displaying, hiding, sizing, or docking the toolbars. Toolbars can be docked to any side of your screen. You can also arrange toolbars by snapping them to the edges of other toolbars or the Property Bar. Snapping makes it easy to arrange toolbars on screen and organize your work area.

{button ,AL(`CLS_Tools')} **Related Topics**

Tools.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

{button ,AL(`CLS_Tools;FNC_Application')} **Related Topics**

Tools.Count

Property **Count** As Long

Description

The **Count** property returns the number of tools in the **Tools** collection of Corel PHOTO-PAINT. The Corel PHOTO-PAINT Toolbox contains tools for creating and manipulating objects and selections in your image. The zoom tools let you view specific areas of your image, and the shaping and painting tools let you modify your image. The Toolbox also contains tools that let you apply modifications interactively.

The **Count** property returns a Read-Only value.

{button ,AL(`CLS_Tools;FNC_Count')} **Related Topics**

Tools.Item

Property **Item**(ByVal **IndexOrName** As Variant) As Object

Description

The **Item** property returns a value associated with the index number or name of a tool object in the **Tools** collection of Corel PHOTO-PAINT. The Corel PHOTO-PAINT Toolbox contains tools for creating and manipulating objects and selections in your image. The zoom tools let you view specific areas of your image, and the shaping and painting tools let you modify your image. The Toolbox also contains tools that let you apply modifications interactively.

The **Item** property returns a Read-Only value. The **Item** property is the default property and may be omitted when referencing items in the collection.

You must reference a tool in the collection by passing its index number or name as a parameter:

Parameters	Description
IndexOrName	**Index** is a preset placeholder for each object in a **Tools** collection; it uniquely identifies each member of the collection. **Name** is the unique text name given to each tool object.

{button ,AL(`CLS_Tools;FNC_Item')} **Related Topics**

Tools.Parent

Property **Parent** As **Document**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

{button ,AL(`CLS_Tools;FNC_Parent')} **Related Topics**

Window properties

Window Legend

- Active

- Application

- Caption

- Document
  - GridVisible
  - GuidesVisible
  - Height

- Index
  - Left

- Next

- Parent

- Previous
  - RulerOriginX
  - RulerOriginY
  - RulersVisible
  - SnapToGrid
  - SnapToGuides
  - Top
  - ViewCenterX
  - ViewCenterY
  - Visible
  - Width
  - WindowState
  - Zoom

Window methods

Window Legend

- Activate
- Close
- NewWindow
- SetViewCenter

Window

Class **Window**

Properties Methods Referenced By

The **Window** class defines the characteristics of Window objects and describes the look and behavior of the objects through its properties and methods.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

{button ,AL(`CLS_Window')} **Related Topics**

Window.Active

Property **Active** As Boolean

Description

The **Active** property returns a True or False value indicating the activity status of a window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

If the window is active, a value of True is returned, indicating that it is currently in use. If the window is inactive, the return value is False, indicating that the window is currently open but not in use.

Use the **Activate** method to change the activity status of the current window.

The **Active** property returns a Read-Only value.

Example

The following code example displays the activity status of the current window in a message box:

```
Sub WindowActive()
MsgBox ActiveWindow.Active
End Sub
```

{button ,AL(`CLS_Window;FNC_Active')} **Related Topics**

Window.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub WindowApplication()
With ActiveWindow.Application
    MsgBox .Version
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Application')} **Related Topics**

Window.Caption

Property **Caption** As String

Description

The **Caption** property returns the caption of the active window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The caption appears as text in the title bar of the main application window. If the window is maximized when using the **Caption** property, the document title also displays after the caption in the title bar.

The **Caption** property returns a Read-Only value.

Example

The following code example displays the caption of the active window in a message box:

```
Sub WindowCaption()
With ActiveWindow
    MsgBox .Caption
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Caption')} **Related Topics**

Window.Document

Property **Document** As **Document**

Definition

The **Document** property returns a value associated with a document contained in the active window of Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

A document is a single page image or a collection of single-page images that make up a multi-page document.

The **Document** property returns a Read-Only value.

Example

The following code example displays the name of the active window's document in a message box:

```
Sub WindowDocument()
With ActiveWindow.Document
    MsgBox .Name
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Document')} **Related Topics**

Window.GridVisible

Property **GridVisible** As Boolean

Description

The **GridVisible** property returns or sets a True or False value that indicates whether or not a grid is visible in a window of Corel PHOTO-PAINT. A Grid is a series of evenly spaced horizontal and vertical dots that are used to help draw and arrange objects. You can use the controls on the Grid and Ruler Setup dialog box to set the grid's parameters. For greater accuracy, you can also have objects in your illustration snap to the grid when you move, or draw them.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example sets the **GridVisible** property to True in the active window:

```
Sub Grid()
With ActiveWindow
    .GridVisible = True
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_GridVisible')} **Related Topics**

Window.GuidesVisible

Property **GuidesVisible** As Boolean

Description

The **GuidesVisible** property returns or sets a True or False value that indicates whether or not the guides are visible in a window of Corel PHOTO-PAINT. A guideline is a line placed anywhere in the Corel PHOTO-PAINT image window that is used to help align and position image objects. Guidelines are used with the rulers and the grid to draw objects with precision.

Guidelines may be Horizontal, Vertical, or Slanted and any number of them may be created and saved with your image. You can also enable snapping guidelines so that objects automatically align with the guidelines when moved or drawn nearby.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example sets the **GuidesVisible** property to True in the active window:

```
Sub Guide()
With ActiveWindow
    .GuidesVisible = True
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_GuidesVisible')} **Related Topics**

Window.Height

Property **Height** As Long

Description

The **Height** property returns or sets the height of the active window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The height of the window is a numerical value measured in screen pixels and measures from the top to the bottom of the window.

Example

The following code example sets the height of the active window to 200 pixels:

```
Sub WindowHeight()
With ActiveWindow
    .Height = 200
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Height')} **Related Topics**

Window.Index

Property **Index** As Long

Description

The **Index** property returns a value associated with a window object in the **Windows** collection of Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Every window in Corel PHOTO-PAINT is identified by an index number. For example, `Windows(2)`, or `Window.Index (2)` both refer to the second window in the **Windows** collection of Corel PHOTO-PAINT. The number 2 is the window's index number.

The **Index** property returns a Read-Only value.

Example

The following code example closes the active window if the index of the active window is 1:

```
Sub WindowIndex()
With ActiveWindow
    If .Index = 1 Then
    .Close
End If
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Index')} **Related Topics**

Window.Left

Property **Left** As Long

Description

The **Left** property sets the location of the left border of the active window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The **Left** property sets the distance, in screen pixels, from the active window's left border to the left side of main application window.

Example

The following code example sets the **Left** property to 100 screen pixels:

```
Sub WindowTop()
With ActiveWindow
    .Left = 100
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Left')} **Related Topics**

Window.Next

Property **Next** As **Window**

Description

The **Next** method returns the window in the **Windows** collection that follows the selected window object in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

For example, if the selected window object has an index value of 2, the **Next** method returns the window with an index value of 3 in the **Windows** collection.

The **Next** property returns a Read-Only value.

Example

The following code example displays the caption of the window that follows the active window in the **Windows** collection:

```
Sub WindowPrevious()
With ActiveWindow
    MsgBox .Next.Caption
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Next')} **Related Topics**

Window.Parent

Property **Parent** As **Windows**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the caption of the window's parent object in a message box:

```
Sub WindowParent()
With ActiveWindow
    MsgBox .ActiveView.Parent.Caption
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Parent')} **Related Topics**

Window.Previous

Property **Previous** As **Window**

Description

The **Previous** method returns the window in the **Windows** collection that precedes the selected window object in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

For example, if the selected window object has an index value of 2, the **Previous** method returns the window with an index value of 1 in the **Windows** collection.

The **Previous** property returns a Read-Only value.

Example

The following code example displays the caption of the window that precedes the active window in the **Windows** collection:

```
Sub WindowPrevious()
With ActiveWindow
    MsgBox .Previous.Caption
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Previous')} **Related Topics**

Window.RulerOriginX

Property **RulerOriginX** As Long

Description

The **RulerOriginX** property returns or sets the origin of the horizontal ruler in Corel PHOTO-PAINT. The origin is the point where the horizontal and vertical rulers meet in the application window. A ruler is a measuring tool that displays on the left side and along the top of the Application Window. The rulers help you size and position the objects in your image.

You can choose when and how you want to display the rulers and the grid. If screen space is limited, you might choose to hide the rulers and display them only when you need them. Alternatively, if you want to view your image as it will appear when you print it, you can hide the grid and display it later. The rulers and grid maintain their settings even when you hide them.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code displays the Y origin value of the horizontal ruler in Corel PHOTO-PAINT:

```
Sub Ruler()
With ActiveWindow
    MsgBox .RulerOriginX
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_RulerOriginX')} **Related Topics**

Window.RulerOriginY

Property **RulerOriginY** As Long

Description

The **RulerOriginY** property returns or sets the origin of the vertical ruler in Corel PHOTO-PAINT. The origin is the point where the horizontal and vertical rulers meet in the application window. A ruler is a measuring tool that displays on the left side and along the top of the Application Window. The rulers help you size and position the objects in your image.

You can choose when and how you want to display the rulers and the grid. If screen space is limited, you might choose to hide the rulers and display them only when you need them. Alternatively, if you want to view your image as it will appear when you print it, you can hide the grid and display it later. The rulers and grid maintain their settings even when you hide them.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code displays the Y origin value of the vertical ruler in Corel PHOTO-PAINT:

```
Sub Ruler()
With ActiveWindow
    MsgBox .RulerOriginY
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_RulerOriginY')} **Related Topics**

Window.RulersVisible

Property **RulersVisible** As Boolean

Description

The **RulersVisible** property returns or sets a True or False value that indicates whether or not the rulers are visible in the active window of Corel PHOTO-PAINT. A ruler is a measuring tool that displays on the left side and along the top of the Application Window. The rulers help you size and position the objects in your image.

You can choose when and how you want to display the rulers and the grid. If screen space is limited, you might choose to hide the rulers and display them only when you need them. Alternatively, if you want to view your image as it will appear when you print it, you can hide the grid and display it later. The rulers and grid maintain their settings even when you hide them.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example sets the **RulersVisible** property to True in the active window:

```
Sub Guide()
With ActiveWindow
    .RulersVisible = True
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_RulersVisible')} **Related Topics**

Window.SnapToGrid

Property **SnapToGrid** As Boolean

Description

The **SnapToGrid** property returns or sets a True or False value that indicates whether or not an object will snap to the grid in a window of Corel PHOTO-PAINT. A Grid is a series of evenly spaced horizontal and vertical dots that are used to help draw and arrange objects. You can use the controls on the Grid and Ruler Setup dialog box to set the grid's parameters. For greater accuracy, you can also have objects in your illustration snap to the grid when you move, or draw them.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example sets the **SnapToGrid** property to True in the active window:

```
Sub Grid()
With ActiveWindow
    .SnapToGrid = True
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_SnapToGrid')} **Related Topics**

Window.SnapToGuides

Property **SnapToGuides** As Boolean

Description

The **SnapToGuides** property returns or sets a True or False value that indicates whether or not an object will snap to the guides in a window of Corel PHOTO-PAINT. A guideline is a line placed anywhere in the Corel PHOTO-PAINT image window that is used to help align and position image objects. Guidelines are used with the rulers and the grid to draw objects with precision.

Guidelines may be Horizontal, Vertical, or Slanted and any number of them may be created and saved with your image. You can also enable snapping guidelines so that objects automatically align with the guidelines when moved or drawn nearby.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example sets the **SnapToGuides** property to True in the active window:

```
Sub Guide()
With ActiveWindow
    .SnapToGuide = True
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_SnapToGuides')} **Related Topics**

Corel PHOTO-PAINT's VBA programming commands that specify locations on a page use coordinates as parameters. Coordinate values use the set document unit as the base unit of measurement and are expressed as being relative to the center of the current page, which has the coordinates (0,0).

cdrTenthMicron=0
cdrInch=1
cdrFoot=2
cdrMillimeter=3
cdrCentimeter=4
cdrPixel=5
cdrMile=6
cdrMeter=7
cdrKilometer=8
cdrDidots=9
cdrAgate=10
cdrYard=11
cdrPica=12
cdrCicero=13
cdrPoint=14

Window.Top

Property **Top** As Long

Description

The **Top** property sets the location of the left border of the active window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The **Top** property sets the distance, in screen pixels, from the active window's top border to the top of the application window's border in Corel PHOTO-PAINT.

Example

The following code example sets the **Top** property to 100 screen pixels:

```
Sub WindowTop()
With ActiveWindow
    .Top = 100
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Top')} **Related Topics**

Window.ViewCenterX

Property **ViewCenterX** As Long

Description

The **ViewCenterX** property returns or sets the horizontal center of display in a window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example displays the horizontal value of the current view center in the active window of Corel PHOTO-PAINT:

```
Sub View()
With ActiveWindow
    MsgBox .ViewCenterX
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_ViewCenterX')} **Related Topics**

Window.ViewCenterY

Property **ViewCenterY** As Long

Description

The **ViewCenterY** property returns or sets the vertical center of display in a window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example displays the vertical value of the current view center in the active window of Corel PHOTO-PAINT:

```
Sub View()
With ActiveWindow
    MsgBox .ViewCenterY
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_ViewCenterY')} **Related Topics**

Window.Visible

Property **Visible** As Boolean

Description

The **Visible** property returns or sets a True or False value indicating the visibility status of Corel PHOTO-PAINT. The application object refers to the Corel PHOTO-PAINT application. Corel PHOTO-PAINT is a vector-based image application that lets you create professional artwork, from simple logos to intricate technical illustrations.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

If the Visible property is True, the application is running and visible on the monitor display. This property allows you to show or hide the Corel PHOTO-PAINT application window.

Example

The following code example sets the visibility of the active window to True:

```
Sub LayerVisible()
ActiveWindow.Visible = True
End Sub
```

{button ,AL(`CLS_Window;FNC_Visible')} **Related Topics**

Window.Width

Property **Width** As Long

Description

The **Width** property returns or sets the width of the active window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The width of the window is a numerical value measured in screen pixels and measures from the left to the right of the window.

Example

The following code example sets the width of the active window to 200 pixels:

```
Sub WindowHeight()
With ActiveWindow
    .Width = 200
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Width')} **Related Topics**

Window.WindowState

Property **WindowState** As **cdrWindowState**

Description

The **WindowState** property returns or sets the window state of the current window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

A window can be minimized, maximized or in a normal state.

The **WindowState** property returns a value of cdrWindowState.

Example

The following code example minimizes the active window of Corel PHOTO-PAINT:

```
Sub WindowState()
With ActiveWindow
    .WindowState = cdrWindowMinimized
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_WindowState')} **Related Topics**

cdrWindowNormal	1
cdrWindowMaximized	3
cdrWindowMinimized	2
cdrWindowRestore	9

Window.Zoom

Property **Zoom** As Long

Description

The **Zoom** property returns a preset, or sets a user-defined magnification level in the active view of Corel PHOTO-PAINT. Changing the magnification level allows you to zoom in and get a closer look at an area of your image, or zoom out and see a wider area of the document. The default zoom level for Corel PHOTO-PAINT is 100%.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example sets the zoom level of the active window to 200%, if the current zoom level is less than 200%:

```
Sub Zoom()
With ActiveWindow
    If .Zoom < 200 Then
    .Zoom = 200
    End If
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Zoom')} **Related Topics**

Window.Activate

Sub **Activate**()

Description

The **Activate** method opens a window in Corel PHOTO-PAINT and makes it the active window, if it is not currently open. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

In may cases, newer operating systems like Windows 98 and Windows 2000 don't allow the application to activate itself, to prevent intervention with you're current activity. If the application is not granted active status in certain situations, the application window will start flashing in the taskbar, but it will not appear in the foreground.

Example

The following code example activates the current window in Corel PHOTO-PAINT, if it is not already open:

```
Sub WindowActivate()
If ActiveWindow.Active = False Then
    ActiveWindow.Activate
End If
End Sub
```

{button ,AL(`CLS_Window;FNC_Activate')} **Related Topics**

Window.Close

Sub **Close**()

Description

The **Close** method closes the active window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an image Window. You can draw anywhere in the window, but only objects that appear on the Image Page will print. This area is enclosed by a rectangle with a shadow effect. Although you can draw anywhere in the Image Window, only objects on the Image Page appear in your print jobs.

Before you close an active file, Corel PHOTO-PAINT prompts you to save the file. You can also close specific windows, as well as close all open files.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example closes the active window:

```
Sub WindowClose()
With ActiveWindow
    .Close
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_Close')} **Related Topics**

Window.NewWindow

Function **NewWindow**() As **Window**

Description

The **NewWindow** method creates a new window in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Example

The following code example creates a new window in Corel PHOTO-PAINT. A message displays the number of open windows, including the new window, in a message box:

```
Sub WindowNew()
ActiveWindow.NewWindow
    MsgBox Windows.Count
End Sub
```

{button ,AL(`CLS_Window;FNC_NewWindow')} **Related Topics**

Window.SetViewCenter

Sub **SetViewCenter**(ByVal **X** As Long, ByVal **Y** As Long)

Description

The **SetViewCenter** method sets the central viewpoint, or center of display, in the active window of Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

Parameters	Description
X	The **X** parameter identifies the horizontal coordinate of the viewpoint in a window. The viewpoint is the center point of a window's view. This value is measured in document units.
Y	The **Y** parameter identifies the vertical coordinate of the viewpoint in a window. The viewpoint is the center point of a window's view. This value is measured in document units.

Example

The following code example sets the center of the viewpoint of the active window with the **SetViewPoint** method at the point (2, 2):

```
Sub Center()
With ActiveWindow
    .SetViewCenter 2, 2
End With
End Sub
```

{button ,AL(`CLS_Window;FNC_SetViewCenter')} **Related Topics**

Windows properties

Windows Legend

- Application
- Count
-
- Item
- Parent

Windows methods

Windows Legend

- Arrange
- CloseAll

Windows

Class **Windows**

Properties Methods Referenced By

The **Windows** class defines the characteristics of **Windows** collection objects and describes the look and behavior of the collection objects through its properties and methods.

A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

{button ,AL(`CLS_Windows')} **Related Topics**

cdrTileHorizontally=0

cdrTileVertically=1

cdrCascade=2

Windows.Application

Property **Application** As **Application**

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub ApplicationWindows()
With Windows
MsgBox .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Windows;FNC_Application')} **Related Topics**

Windows.Count

Property **Count** As Long

Description

The **Count** property returns the number of windows in the **Windows** collection of Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

The **Count** property returns a Read-Only value.

Example

The following code example displays the number of windows in **Windows** collection of Corel PHOTO-PAINT:

```
Sub WindowsCount()
MsgBox Windows.Count
End Sub
```

{button ,AL(`CLS_Windows;FNC_Count')} **Related Topics**

Windows.Item

Property **Item**(ByVal **Index** As Long) As **Window**

Description

The **Item** property returns a value associated with the index number of a window object in the **Windows** collection of Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

`Windows(2)` refers to the second window in the **Windows** collection of Corel PHOTO-PAINT.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Windows.Item(2)` is the same as `Windows(2)`. Both structures reference the second window in the **Windows** collection.

You must reference a window in the collection by passing its index number as a parameter:

Parameters	Description
Index	**Index** is a preset placeholder for each object in a Windows collection; it uniquely identifies each member of the collection.

Example

The following code example displays the caption of the second window in the **Windows** collection in a message box:

```
Sub WindowsItem()
MsgBox Windows.Item(2).Caption
End Sub
```

{button ,AL(`CLS_Windows;FNC_Item')} **Related Topics**

Windows.Parent

Property **Parent** As **Application**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the path of the window collection's parent object in a message box:

```
Sub ParentWindows()
With Windows
MsgBox .Parent.Path
End With
End Sub
```

{button ,AL(`CLS_Windows;FNC_Parent')} **Related Topics**

Windows.Arrange

Sub **Arrange**(ByVal **Style** As **cdrWindowArrangeStyle**)

Description

The **Arrange** method organizes all open windows in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

A window can be tiled horizontally or vertically, as well as cascaded in the application window.

You must reference the style in which to organize the windows by passing the style type as a parameter:

Parameters	Description
Style	The **Style** parameter identifies the type of window arrangement in Corel PHOTO-PAINT. The styles for windows in Corel PHOTO-PAINT are Tile Vertically, Tile Horizontally, and Cascade.

Example

The following code example uses the **Arrange** method to cascade all open windows in Corel PHOTO-PAINT:

```
Sub WindowsArrange()
Windows.Arrange (cdrCascade)
End Sub
```

{button ,AL(`CLS_Windows;FNC_Arrange')} **Related Topics**

Windows.CloseAll

Sub **CloseAll**()

Description

The **CloseAll** method closes all open windows in Corel PHOTO-PAINT. A window is a container for an image in Corel PHOTO-PAINT and is often referred to as an Image Window. An image that you open or create in Corel PHOTO-PAINT appears in an Image Window. You can open as many images as your computer's memory permits. If you have more than one image open, you can click inside an Image Window to make that image active. You can move an Image Window by dragging its Title Bar.

You are prompted to save your changes in each window when you execute the **CloseAll** method.

Example

The following code example uses the **CloseAll** method to exit all the open windows in Corel PHOTO-PAINT:

```
Sub WindowsClose()
Windows.CloseAll
End Sub
```

{button ,AL(`CLS_Windows;FNC_CloseAll')} **Related Topics**

Workspace properties

Workspace Legend

- Active
- Application
- Default
- Description
- Name
- Parent

Workspace methods

Workspace Legend

Activate

Workspace

Class **Workspace**

Properties Methods Referenced By

The **Workspace** class defines the characteristics of workspace objects and describes the look and behavior of the objects through its properties and methods.

A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

You can create multiple workspaces for specific users or specific tasks and then apply them when required. You can also delete workspaces when they are no longer needed.

You can customize your workspace settings. You can set up your screen the way you want, choose options in the Options dialog box, and then create a custom workspace to save your settings. You can customize the tools and operations that you use most, such as menus and shortcut keys. You can access your custom settings by loading your saved workspace.

{button ,AL(`CLS_Workspace')} **Related Topics**

Workspace.Active

Property **Active** As Boolean

Description

The **Active** property returns a True or False value indicating if a specified workspace is currently active in Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

If the **Active** property is True, the workspace is currently active in the application.

The **Active** property returns a Read-Only value.

Example

The following code example displays the name of the current workspace, if the workspace is active, in a message box:

```
Sub WorkspaceActive()
If ActiveWorkspace.Active = True Then
    MsgBox ActiveWorkspace.Name
End If
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Active')} **Related Topics**

Workspace.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub WorkspaceApplication()
With ActiveWorkspace
    MsgBox .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Application')} **Related Topics**

Workspace.Default

Property **Default** As Boolean

Description

The **Default** property returns a True or False value indicating if the current workspace is the default workspace in Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

If the default property is True, the current workspace is the default workspace. The default workspace can be set in the Options dialog box.

The **Default** property returns a Read-Only value.

Example

The following code example checks to see if the current workspace is the default workspace. If it is NOT the default workspace, a message box displays the name of the active workspace in a message box:

```
Sub WorkspaceDefault()
If ActiveWorkspace.Default = False Then
    MsgBox ActiveWorkspace.Name
End If
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Default')} **Related Topics**

Workspace.Description

Property **Description** As String

Description

The **Description** property returns a written description of a workspace in Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

When a workspace is created, you can provide a brief description of the workspace. This is useful if you are using multiple workspaces. The description that you type in the Description Of New Workspace box appears in the list of available workspaces.

The **Description** property returns a Read-Only value.

Example

The following code example displays the description text of the current workspace in a message box:

```
Sub WorkspaceDescription()
MsgBox ActiveWorkspace.Description
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Description')} **Related Topics**

Workspace.Name

Property **Name** As String

Description

The **Name** property returns the name given to a workspace in Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

The **Name** property returns a String value.

Example

The following code example displays the name of the current workspace in a message box:

```
Sub WorkspaceName()
MsgBox ActiveWorkspace.Name
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Name')} **Related Topics**

Workspace.Parent

Property **Parent** As **Workspaces**

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the path of the workspace's parent object in a message box:

```
Sub WorkspaceParent()
With ActiveWorkspace
    MsgBox .Parent.Path
End With
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Parent')} **Related Topics**

Workspace.Activate

Sub **Activate**()

Description

The **Activate** method opens a workspace, if the workspace is not currently open in Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

Example

The following code example uses the **Activate** method to open the current workspace if it is not already open in Corel PHOTO-PAINT:

```
Sub WorkspaceActivate()
If Not ActiveWorkspace.Active Then
    ActiveWorkspace.Activate
End If
End Sub
```

{button ,AL(`CLS_Workspace;FNC_Activate')} **Related Topics**

Workspaces properties

Workspaces Legend

- Application

- Count

- Item

- Parent

Workspaces

Class **Workspaces**

Properties Referenced By

The **Workspaces** class defines the characteristics of **Workspaces** collection objects and describes the look and behavior of the collection objects through its properties and methods. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT.

A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements. For example, if you are using a low-resolution monitor setting, you can use the preset workspace designed for such a setting.

You can create multiple workspaces for specific users or specific tasks and then apply them when required. You can also delete workspaces when they are no longer needed.

You can customize your workspace settings. You can set up your screen the way you want, choose options in the Options dialog box, and then create a custom workspace to save your settings. You can customize the tools and operations that you use most, such as menus and shortcut keys. You can access your custom settings by loading your saved workspace.

{button ,AL(`CLS_Workspaces')} **Related Topics**

Workspaces.Application

Property **Application** As Object

Description

The **Application** property returns a value associated with the main Corel PHOTO-PAINT application. An application object is always available in VBA and its properties and methods are global, making it unnecessary to explicitly specify the Application object. For example, with the Application object, CorelScript and Application.CorelScript can be used interchangeably because they represent the same application object.

The **Application** property returns a Read-Only value.

Example

The following code example displays the current version of Corel PHOTO-PAINT in a message box:

```
Sub WorkspacesApplication()
With Workspaces
    MsgBox .Application.Version
End With
End Sub
```

{button ,AL(`CLS_Workspaces;FNC_Application')} **Related Topics**

Workspaces.Count

Property **Count** As Long

Description

The **Count** property returns the number of workspaces in the **Workspaces** collection of Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements.

The **Count** property returns a Read-Only value.

Example

The following code example displays the number of available workspace objects in the **Workspaces** collection of Corel PHOTO-PAINT in a message box:

```
Sub WorkspacesCount()
MsgBox Workspaces.Count
End Sub
```

{button ,AL(`CLS_Workspaces;FNC_Count')} **Related Topics**

Workspaces.Item

Property **Item**(ByVal **IndexOrName** As Variant) As **Workspace**

Description

The **Item** property returns a value associated with the index number or name of a workspace object in the **Workspaces** collection of Corel PHOTO-PAINT. A workspace is an enclosed work environment that is ideally suited to meet specific needs when creating an image in Corel PHOTO-PAINT. You can choose from several preset workspaces. Each preset workspace is designed to provide a working environment tailored to your requirements.

A workspace in Corel PHOTO-PAINT is identified by an index number or by its name. You can view the name of each workspace is the Options dialog box.

`Workspaces(2)` refers to the second workspace in the **Workspaces** collection of Corel PHOTO-PAINT. If this workspace is called "Enhanced", `Workspaces(Enhanced)` also refers to the second workspace in the **Workspaces** collection.

The **Item** property returns a Read-Only value.

The **Item** property is the default property and may be omitted when referencing items in the collection. For example, `Workspaces.Item(2)` is the same as `Workspaces(2)`. They both reference the second workspace in the **Workspaces** collection.

You must reference a workspace in the collection by passing its index number or name as a parameter:

Parameters	Description
IndexOrName	**Index** is a preset placeholder for each object in a **Workspaces** collection; it uniquely identifies each member of the collection. **Name** is the unique text name given to each workspace. Workspace names can be viewed in the Options dialog box.

{button ,AL(`CLS_Workspaces;FNC_Item')} **Related Topics**

Workspaces.Parent

Property **Parent** As Object

Description

The **Parent** property returns a value associated with the properties, methods, and controls of an object's parent in Corel PHOTO-PAINT's hierarchy of objects. A parent object acts as a container for child objects. For example, a form, as a container, is the parent object of a check box on that form.

Most objects have either a **Parent** property or a **Collection** property, pointing to the object's parent object in the object model. The **Collection** property is used if the parent object is a collection.

The **Parent** property returns a Read-Only value.

Example

The following code example displays the path of the **Workspaces** collection's parent object in a message box:

```
Sub WorkspacesParent()
With Workspaces
MsgBox .Parent.Path
End With
End Sub
```

{button ,AL(`CLS_Workspaces;FNC_Parent')} **Related Topics**

cdrColorType

Enum **cdrColorType**

Referenced By

Constant	Value	Description
cdrColorPantone	1	
cdrColorCMYK	2	
cdrColorCMY	4	
cdrColorRGB	5	
cdrColorHSB	6	
cdrColorHLS	7	
cdrColorBlackAndWhite	8	
cdrColorGray	9	
cdrColorYIQ	11	
cdrColorLab	12	
cdrColorPantoneHex	14	
cdrColorRegistration	20	
cdrColorSpot	25	
cdrColorMixed	99	

{button ,AL(`CLS_cdrColorType')} **Related Topics**

cdrDitherType

Enum **cdrDitherType**

Referenced By

Constant	Value	Description
cdrDitherNone	0	
cdrDitherOrdered	1	
cdrDitherJarvis	2	
cdrDitherStucki	3	
cdrDitherFloyd	4	

{button ,AL(`CLS_cdrDitherType')} **Related Topics**

cdrDuotoneType

Enum **cdrDuotoneType**

Referenced By

Constant	Value	Description
cdrMonotone	0	
cdrDuotone	1	
cdrTritone	2	
cdrQuadtone	3	

{button ,AL(`CLS_cdrDuotoneType')} **Related Topics**

cdrFilter

Enum **cdrFilter**

Referenced By

Constant	Value	Description
cdrAutoSense	0	
cdrBMP	769	
cdrPCX	770	
cdrTGA	771	
cdrTIFF	772	
cdrGIF	773	
cdrJPEG	774	
cdrPCD	775	
cdrPSD	788	
cdrPP	789	
cdrPIC	790	
cdrMAC	791	
cdrOS2BMP	792	
cdrEPSPhotoPaint	804	
cdrRAWPhotoPaint	805	
cdrFPX	806	
cdrRIFF	807	
cdrPICT4	808	
cdrWPG	1287	
cdrPSEncapsulated	1289	
cdrPSInterpreted	1290	
cdrOS2Metafile	1291	
cdrCMF	1295	
cdrDXF	1296	
cdrRND	1297	
cdrTTF	1302	
cdrAT1	1303	
cdrWPG2	1304	
cdrBarista	1312	
cdrPDFPlaceable	1314	
cdrVSD	1315	
cdrPICTWithEPS	1316	
cdrDWG	1328	
cdrAVI	1536	
cdrCMV	1539	
cdrSHW	1540	

cdrCCH	1541
cdrQTM	1542
cdrCFL	1550
cdrMPEG	1551
cdrQTVR	1560
cdrCPT	1792
cdrCMX6	1793
cdrCMX5	1794
cdrCDR	1795
cdrCDX	1796
cdrCPX	1797
cdrCCD	1798
cdrCPT7	1799
cdrCDT	1800
cdrPAT	1801
cdrCPT9	1808
cdrCPH	1803
cdrSWF	1343
cdrSVG	1345
cdrFH	1344
cdrSCT	776
cdrIMG	787
cdrPNG	802
cdrCGM	1280
cdrPLT	1281
cdrDRW	1282
cdrGEM	1284
cdrPIF	1285
cdrPICT	1288
cdrWMF	1294
cdrSCD	1298
cdrEMF	1300
cdrEPS	1289
cdrWI	793
cdrXPM	809
cdrWVL	793
cdrAI	1305
cdrHPGL	1281
cdrCPT10	1808
cdrCAL	800
cdrCLK	1802
cdr3DMF	1559

cdrTXT	2048
cdrRTF	2053
cdrDOC	2068
cdrWord2000	2068
cdrWord95	2049
cdrWord2	2050
cdrWord55	2051
cdrMacWord5	2052
cdrWP4	2058
cdrWP50	2057
cdrWP51	2056
cdrWP9	2055
cdrWPD	2055
cdrWSW	2059
cdrWSD	2061
cdrWordStar2000	2061
cdrWordStar7	2060
cdrXY	2062
cdrSAM	2063
cdrXLS	2065
cdrLotus123	2066
cdrWK	2066
cdrWPM	2072
cdrICO	784
cdrCUR	785
cdrEXE	786
cdrXCF	816
cdrPNM	817
cdrPP4	789
cdrPP5	803
cdrFMV	1329
cdrMET	1291
cdrNAP	1292
cdrGIFAnimation	1558

{button ,AL(`CLS_cdrFilter')} **Related Topics**

cdrFlipAxes

Enum **cdrFlipAxes**

Referenced By

Constant	Value	Description
cdrFlipHorizontal	1	
cdrFlipVertical	2	
cdrFlipBoth	3	

{button ,AL(`CLS_cdrFlipAxes')} **Related Topics**

cdrHalftoneType

Enum **cdrHalftoneType**

Referenced By

Constant	Value	Description
cdrHalftoneSquare	0	
cdrHalftoneRound	1	
cdrHalftoneLine	2	
cdrHalftoneCross	3	
cdrHalftoneFixed4x4	4	
cdrHalftoneFixed8x8	5	

{button ,AL(`CLS_cdrHalftoneType')} **Related Topics**

cdrImageMode

Enum **cdrImageMode**

Referenced By

Constant	Value	Description
cdrImageRGB	0	
cdrImageGrayscale	1	
cdrImageBlackWhite	2	
cdrImagePaletted	3	
cdrImageCMYK	4	
cdrImageDuotone	5	
cdrImageLAB	6	
cdrImageGrayscale16	7	
cdrImageRGB48	8	
cdrImageMultiChannel	9	

{button ,AL(`CLS_cdrImageMode')} **Related Topics**

cdrImagePaletteType

Enum **cdrImagePaletteType**

Referenced By

Constant	Value	Description
cdrPaletteUniform	0	
cdrPaletteStdVGA	1	
cdrPaletteAdaptive	2	
cdrPaletteOptimized	3	
cdrPaletteBlackBody	4	
cdrPaletteGrayscale	5	
cdrPaletteSystem	6	
cdrPaletteIE	7	
cdrPaletteNetscape	8	
cdrPaletteCustom	9	

{button ,AL(`CLS_cdrImagePaletteType')} **Related Topics**

cdrPaletteID

Enum **cdrPaletteID**

Referenced By

Constant	Value	Description
cdrTRUMATCH	1	
cdrPANTONEProcess	2	
cdrPANTONECorel8	3	
cdrUniform	7	
cdrFOCOLTONE	8	
cdrSpectraMaster	9	
cdrTOYO	10	
cdrDIC	11	
cdrPANTONEHexCoated	12	
cdrPANTONEHexUncoated	24	
cdrLab	13	
cdrNetscapeNavigator	14	
cdrInternetExplorer	15	
cdrPANTONECoated	17	
cdrPANTONEUncoated	18	
cdrPANTONEMetallic	20	
cdrPANTONEPastelCoated	21	
cdrPANTONEPastelUncoated	22	
cdrHKS	23	
cdrCustom	0	

{button ,AL(`CLS_cdrPaletteID')} **Related Topics**

cdrPaletteSortMethod

Enum **cdrPaletteSortMethod**

Referenced By

Constant	Value	Description
cdrSortReverse	0	
cdrSortHue	1	
cdrSortBrightness	2	
cdrSortSaturation	3	
cdrSortRGB	4	
cdrSortHSB	5	
cdrSortName	6	

{button ,AL(`CLS_cdrPaletteSortMethod')} **Related Topics**

cdrPaletteType

Enum **cdrPaletteType**

Referenced By

Constant	Value	Description
cdrFixedPalette	0	
cdrCustomPalette	1	

{button ,AL(`CLS_cdrPaletteType')} **Related Topics**

cdrRenderType

Enum **cdrRenderType**

Referenced By

Constant	Value	Description
cdrRenderLineArt	0	
cdrRenderOrdered	1	
cdrRenderJarvis	2	
cdrRenderStucki	3	
cdrRenderFloyd	4	
cdrRenderHalftone	5	
cdrRenderCardinality	6	

{button ,AL(`CLS_cdrRenderType')} **Related Topics**

cdrWindowArrangeStyle

Enum **cdrWindowArrangeStyle**

Referenced By

Constant	Value	Description
cdrTileHorizontally	0	
cdrTileVertically	1	
cdrCascade	2	

{button ,AL(`CLS_cdrWindowArrangeStyle')} **Related Topics**

cdrWindowState

Enum **cdrWindowState**

Referenced By

Constant	Value	Description
cdrWindowNormal	1	
cdrWindowMaximized	3	
cdrWindowMinimized	2	
cdrWindowRestore	9	

{button ,AL(`CLS_cdrWindowState')} **Related Topics**

pntAddinFilter

Enum **pntAddinFilter**

Referenced By

Constant	Value	Description
pntAddinFilterNone	0	
pntAddinFilterWhileImage	1	
pntAddinFilterShapeCreated	2	
pntAddinFilterNew	4	
pntAddinFilterExecute	8	

{button ,AL(`CLS_pntAddinFilter')} **Related Topics**

pntAlignType

Enum **pntAlignType**

Referenced By

Constant	Value	Description
pntAlignLeft	1	
pntAlignRight	2	
pntAlignHCenter	3	
pntAlignTop	4	
pntAlignBottom	8	
pntAlignVCenter	12	

{button ,AL(`CLS_pntAlignType')} **Related Topics**

pntBorderType

Enum **pntBorderType**

Referenced By

Constant	Value	Description
pntBorderHard	0	
pntBorderMedium	1	
pntBorderSoft	2	

{button ,AL(`CLS_pntBorderType')} **Related Topics**

pntClipMaskMode

Enum **pntClipMaskMode**

Referenced By

Constant	Value	Description
pntFromMask	0	
pntFromInvertedMask	1	
pntFromObjectTransparency	2	
pntToShowAll	3	
pntToHideAll	4	
pntUseOpacity	5	

{button ,AL(`CLS_pntClipMaskMode')} **Related Topics**

pntColorMaskMode

Enum **pntColorMaskMode**

Referenced By

Constant	Value	Description
pntColorMaskNormal	0	
pntColorMaskHue	1	
pntColorMaskSaturation	2	
pntColorMaskBrightness	4	

{button ,AL(`CLS_pntColorMaskMode')} **Related Topics**

pntColorMaskType

Enum **pntColorMaskType**

Referenced By

Constant	Value	Description
pntColorMaskSampledColors	0	
pntColorMaskReds	1	
pntColorMaskYellows	2	
pntColorMaskGreens	3	
pntColorMaskCyans	4	
pntColorMaskBlues	5	
pntColorMaskMagentas	6	
pntColorMaskHighlights	7	
pntColorMaskMidtones	8	
pntColorMaskShadows	9	
pntColorMaskOutOfGamut	10	

{button ,AL(`CLS_pntColorMaskType')} **Related Topics**

pntCreationMode

Enum **pntCreationMode**

Referenced By

Constant	Value	Description
pntEmptyLayer	0	
pntFromBackground	1	
pntCopySelection	2	
pntCutSelection	3	

{button ,AL(`CLS_pntCreationMode')} **Related Topics**

pntDistributeType

Enum **pntDistributeType**

Referenced By

Constant	Value	Description
pntDistributeLeft	1	
pntDistributeRight	2	
pntDistributeHCenter	3	
pntDistributeHSpacing	4	
pntDistributeTop	8	
pntDistributeBottom	16	
pntDistributeVCenter	24	
pntDistributeVSpacing	32	

{button ,AL(`CLS_pntDistributeType')} **Related Topics**

pntDropShadowType

Enum **pntDropShadowType**

Referenced By

Constant	Value	Description
pntDropShadowFlat	0	
pntDropShadowBottom	1	
pntDropShadowTop	2	
pntDropShadowLeft	3	
pntDropShadowRight	4	

{button ,AL(`CLS_pntDropShadowType')} **Related Topics**

pntEdgeType

Enum **pntEdgeType**

Referenced By

Constant	Value	Description
pntEdgeLinear	0	
pntEdgeSquared	1	
pntEdgeFlat	2	
pntEdgeInverseSquared	3	
pntEdgeMesa	4	
pntEdgeGaussian	5	

{button ,AL(`CLS_pntEdgeType')} **Related Topics**

pntFeatherType

Enum **pntFeatherType**

Referenced By

Constant	Value	Description
pntFeatherAverage	0	
pntFeatherMiddle	1	
pntFeatherOutside	2	
pntFeatherInside	3	

{button ,AL(`CLS_pntFeatherType')} **Related Topics**

pntFrameBackground

Enum **pntFrameBackground**

Referenced By

Constant	Value	Description
pntFrameFill	0	
pntFrameCopy	1	
pntFrameCopyActive	2	

{button ,AL(`CLS_pntFrameBackground')} **Related Topics**

pntFrameReference

Enum **pntFrameReference**

Referenced By

Constant	Value	Description
pntFirstFrame	0	
pntLastFrame	1	
pntNextFrame	2	
pntPreviousFrame	3	

{button ,AL(`CLS_pntFrameReference')} **Related Topics**

pntGuideType

Enum **pntGuideType**

Referenced By

Constant	Value	Description
pntGuideVertical	0	
pntGuideHorizontal	1	

{button ,AL(`CLS_pntGuideType')} **Related Topics**

pntLayerType

Enum **pntLayerType**

Referenced By

Constant	Value	Description
pntObjectLayer	0	
pntLensLayer	1	

{button ,AL(`CLS_pntLayerType')} **Related Topics**

pntLoadMode

Enum **pntLoadMode**

Referenced By

Constant	Value	Description
pntLoadAll	0	
pntLoadCrop	1	
pntLoadResample	2	

{button ,AL(`CLS_pntLoadMode')} **Related Topics**

pntMaskBrightnessModel

Enum **pntMaskBrightnessModel**

Referenced By

Constant	Value	Description
pntMaskToWhite	0	
pntMaskToBlack	1	

{button ,AL(`CLS_pntMaskBrightnessModel')} **Related Topics**

pntMaskMode

Enum **pntMaskMode**

Referenced By

Constant	Value	Description
pntMaskNormal	0	
pntMaskAdd	1	
pntMaskSubtract	2	
pntMaskXOR	3	

{button ,AL(`CLS_pntMaskMode')} **Related Topics**

pntMatteType

Enum **pntMatteType**

Referenced By

Constant	Value	Description
pntBlackMatte	0	
pntWhiteMatte	1	

{button ,AL(`CLS_pntMatteType')} **Related Topics**

pntMergeMode

Enum **pntMergeMode**

Referenced By

Constant	Value	Description
pntMergeNormal	0	
pntMergeAdd	1	
pntMergeSubtract	2	
pntMergeDifference	3	
pntMergeMultiply	4	
pntMergeDivide	5	
pntMergeIfLighter	6	
pntMergeIfDarker	7	
pntMergeTexturize	8	
pntMergeInvert	9	
pntMergeAND	10	
pntMergeOR	11	
pntMergeXOR	12	
pntMergeBehind	13	
pntMergeScreen	14	
pntMergeOverlay	15	
pntMergeSoftLight	16	
pntMergeHardLight	17	
pntMergeColorDodge	18	
pntMergeColorBurn	19	
pntMergeExclusion	20	

{button ,AL(`CLS_pntMergeMode')} **Related Topics**

pntSplitMode

Enum **pntSplitMode**

Referenced By

Constant	Value	Description
pntSplitRGB	0	
pntSplitCMYK	1	
pntSplitHSB	2	
pntSplitHLS	3	
pntSplitYIQ	4	
pntSplitLAB	5	

{button ,AL(`CLS_pntSplitMode')} **Related Topics**

pntToleranceMode

Enum **pntToleranceMode**

Referenced By

Constant	Value	Description
pntToleranceNormal	0	
pntToleranceHSB	1	

{button ,AL(`CLS_pntToleranceMode')} **Related Topics**

pntURLRegion

Enum **pntURLRegion**

Referenced By

Constant	Value	Description
pntRegionRectangle	0	
pntRegionPolygon	1	
pntRegionCircle	2	

{button ,AL(`CLS_pntURLRegion')} **Related Topics**

Active Property

Select one of the available subtopics below to see detailed help on **Active** property

Object	Property description
AppWindow	Gets the Corel PHOTO-PAINT's window active status
Background	Make background editable
Channel	Gets the active status of the channel.
Layer	ACTIVE
Window	Gets whether this window is the active one.
Workspace	Gets the current workspace

ActiveFrame Property

Select one of the available subtopics below to see detailed help on **ActiveFrame** property

Object	Property description
Application	Gets the reference to the currently active frame
Document	Gets the reference to the currently active frame

ActiveLayer Property

Select one of the available subtopics below to see detailed help on **ActiveLayer** property

Object	Property description
Application	Gets the reference to the currently active layer
Document	Gets the reference to the currently active layer

ActiveWindow Property

Select one of the available subtopics below to see detailed help on **ActiveWindow** property

Object	Property description
Application	Gets the reference to the currently active window
Document	Gets the reference to the currently active window

Application Property

Select one of the available subtopics below to see detailed help on **Application** property

Object	Property description
AddinHook	
AddIns	
Application	Gets the application to which the object belongs
AppWindow	Gets the application to which the object belongs
Background	Gets the application to which the object belongs.
Channel	Gets the application to which the object belongs.
Channels	Gets the application object.
Clipboard	Gets the application to which the object belongs
ClipMask	Gets the application to which the object belongs.
Color	Gets the application to which the object belongs
ColorMask	Gets the application to which the object belongs.
ColorMaskColor	Gets the application to which the object belongs.
ColorMaskColors	Gets the application to which the object belongs.
Colors	Gets the application to which the color collection belongs
ColorTable	Gets the application to which the object belongs.
CorelScriptFile	Gets the application to which the object belongs
Document	Gets the application to which the object belongs
Documents	Gets the application object
DropShadow	Gets the application to which the object belongs.
Floater	Gets the application object.
Frame	Gets the application to which the object belongs.
Frames	Gets the application to which the object belongs.
Guide	Gets the application to which the object belongs.
Guides	Gets the application to which the object belongs.
Layer	Gets the application to which the object belongs.
LayerRange	Gets the application to which the object belongs.
Layers	Gets the application to which the object belongs.
Mask	Gets the application to which the object belongs.
ObjectLayer	Gets the application to which the object belongs.
Palette	Gets the application to which the object belongs
Palettes	Gets the application to which the palette collection belongs
Tools	Gets the application to which the object belongs.
Window	Gets the application object.
Windows	Gets the application object.
Workspace	Gets the application to which the object belongs
Workspaces	Gets the application to which the workspace collection belongs

Background Property

Select one of the available subtopics below to see detailed help on **Background** property

Object	Property description
Document	Gets the background object
Frame	BACKGROUND

Caption Property

Select one of the available subtopics below to see detailed help on **Caption** property

Object	Property description
AppWindow	Gets or sets the Corel PHOTO-PAINT's window caption
Window	Gets the caption of the window.

Color Property

Select one of the available subtopics below to see detailed help on **Color** property

Object	Property description
Channel	Gets or sets the overlay color of the channel.
ColorMaskColor	COLOR
DropShadow	Gets the dropshadow's color.
Palette	Gets or sets a color object based on index

Count Property

Select one of the available subtopics below to see detailed help on **Count** property

Object	Property description
Channels	Gets the number of channels.
ColorMaskColors	COUNT
Colors	Gets the number of colors in the color collection
ColorTable	Gets the number of palettes.
Documents	Gets the number of objects
Frames	Gets the number of frames.
Guides	Gets the number of guides.
LayerRange	COUNT
Layers	Gets the number of layers
Palettes	Gets the number of items in the collection of palettes
Tools	Gets the number of tools available.
Windows	Gets the number of windows.
Workspaces	Gets the number of items in the collection of workspaces

FileName Property

Select one of the available subtopics below to see detailed help on **FileName** property

Object	Property description
CorelScriptFile	Gets the Corel SCRIPT file name
Document	Gets the file name of the document

Height Property

Select one of the available subtopics below to see detailed help on **Height** property

Object	Property description
AppWindow	Gets or sets the Corel PHOTO-PAINT's window height
Window	Gets or sets the height of the window.

Index Property

Select one of the available subtopics below to see detailed help on **Index** property

Object	Property description
AddinHook	
Channel	Gets or sets the index of the channel.
Frame	Gets the index of this frame.
Guide	Gets the index of the guide within the collection
Window	Gets the index of the window.

Item Property

Select one of the available subtopics below to see detailed help on **Item** property

Object	Property description
Channels	Gets the channel object.
ColorMaskColors	ITEM
Colors	Gets a reference to the specified color, in base 1
ColorTable	ITEM
Documents	Gets the document object
Frames	Gets the frame object.
Guides	Gets the guide object.
LayerRange	ITEM
Layers	Gets the layer object.
Palettes	Gets a reference to a specified palette, in base 1
Tools	Gets the given tool object.
Windows	Gets the window object.
Workspaces	Gets a reference to a specified workspace, in base 1

Left Property

Select one of the available subtopics below to see detailed help on **Left** property

Object	Property description
AppWindow	Gets or sets the Corel PHOTO-PAINT's window left position
Window	Gets or sets the position of the window's left border.

Mode Property

Select one of the available subtopics below to see detailed help on **Mode** property

Object	Property description
ColorMask	MODE
Document	Gets the PHOTO-PAINT image mode of the document

Name Property

Select one of the available subtopics below to see detailed help on **Name** property

Object	Property description
Channel	Gets or sets the name of the channel.
Color	Gets color name
CorelScriptFile	Gets or sets a friendly name for the script; will be used as a macro name if the script is translated to VBA
Document	Gets or sets the name of the VBA project corresponding to the document
Layer	Gets the name the specified layers.
Palette	Gets or sets the color palette name; palette name can only be set if it is a custom palette
Workspace	Gets the workspace's name

Opacity Property

Select one of the available subtopics below to see detailed help on **Opacity** property

Object	Property description
Channel	Gets or sets the overlay opacity of the channel.
DropShadow	Gets the dropshadow's opacity.
Layer	Sets the opacity of the selected layer(s).

PaletteID Property

Select one of the available subtopics below to see detailed help on **PaletteID** property

Object	Property description
Color	Gets the Fixed Palette Color's ID
Palette	Gets the palette ID

Parent Property

Select one of the available subtopics below to see detailed help on **Parent** property

Object	Property description
AddinHook	
AddIns	
Application	Gets the application to which the object belongs
AppWindow	Gets the parent of which the object is a child
Background	Gets the parent document object.
Channel	Gets the parent object.
Channels	Gets the parent document object.
Clipboard	Gets the parent of which this object is a child
ClipMask	Gets the parent object.
Color	Gets the parent of which this object is a child
ColorMask	Gets the parent application object.
ColorMaskColor	Gets the parent application object.
ColorMaskColors	Gets the parent application object.
Colors	Gets the parent of which the color collection is a child; if the color object is not from a palette the color parent is null
ColorTable	Gets the parent document object.
CorelScriptFile	Gets the parent of which the object is a child
Document	Gets the application to which the object belongs
Documents	Gets the application object
DropShadow	Gets the parent object.
Floater	Gets the parent document object.
Frame	Gets the parent frame collection.
Frames	Gets the parent document object.
Guide	Gets the parent document object.
Guides	Gets the parent document object.
Layer	Gets the parent document object.
LayerRange	Gets the parent document object.
Layers	Gets the parent document object.
Mask	Gets the parent document object.
ObjectLayer	Gets the parent document object.
Palette	Gets the parent of which this object is a child
Palettes	Gets the parent of which the palette collection is a child
Tools	Gets the parent document object.
Window	Gets the parent window collection.
Windows	Gets the parent document object.
Workspace	Gets the parent of which this object is a child
Workspaces	Gets the parent of which the workspace collection is a child

PositionX Property

Select one of the available subtopics below to see detailed help on **PositionX** property

Object	Property description
Floater	Gets or sets the horizontal position of the floater.
Guide	Gets or sets the horizontal position of the guide
Layer	POSITIONX
LayerRange	POSITIONX
Mask	POSITIONX

PositionY Property

Select one of the available subtopics below to see detailed help on **PositionY** property

Object	Property description
Floater	Gets or sets the vertical position of the floater.
Guide	Gets or sets the vertical position of the guide
Layer	POSITIONY
LayerRange	POSITIONY
Mask	POSITIONY

Selected Property

Select one of the available subtopics below to see detailed help on **Selected** property

Object	Property description
Guide	Gets or sets the selection status of the guide
Layer	Gets the selected layer(s).

SizeHeight Property

Select one of the available subtopics below to see detailed help on **SizeHeight** property

Object	Property description
Document	Gets the height of the document in pixels
Floater	Gets the eight of the floater.
Layer	SIZEHEIGHT
LayerRange	SIZEHEIGHT
Mask	SIZEHEIGHT

SizeWidth Property

Select one of the available subtopics below to see detailed help on **SizeWidth** property

Object	Property description
Document	Gets the width of the document in pixels
Floater	Gets the width of the floater.
Layer	SIZEWIDTH
LayerRange	SIZEWIDTH
Mask	SIZEWIDTH

Top Property

Select one of the available subtopics below to see detailed help on **Top** property

Object	Property description
AppWindow	Gets or sets the Corel PHOTO-PAINT's window top position
Window	Gets or sets the position of the window's top border.

Type Property

Select one of the available subtopics below to see detailed help on **Type** property

Object	Property description
Color	Gets color type
DropShadow	Gets the dropshadow's type.
Guide	Gets the type of the guide
Layer	TYPE
Palette	Gets whether or not a color palette is custom or fixed

Visible Property

Select one of the available subtopics below to see detailed help on **Visible** property

Object	Property description
Application	Returns the application's visibility status; allows you to hide or show the application window
Channel	Gets or sets the visible state of the channel.
ClipMask	Gets or sets the visible state of the clipmask.
Layer	Gets the visible layers.
Window	Gets or sets whether the window is visible.

Width Property

Select one of the available subtopics below to see detailed help on **Width** property

Object	Property description
AppWindow	Gets or sets the Corel PHOTO-PAINT's window width
Window	Gets or sets the width of the window.

Windows Property

Select one of the available subtopics below to see detailed help on **Windows** property

Object	Property description
Application	Gets a collection of windows open in the application
Document	Gets a collection of the document's windows

WindowState Property

Select one of the available subtopics below to see detailed help on **WindowState** property

Object	Property description
AppWindow	Gets or sets the Corel PHOTO-PAINT's window state
Window	Gets or sets the state of the window.

Activate Method

Select one of the available subtopics below to see detailed help on **Activate** method

Object	Method description
AppWindow	Activates Corel PHOTO-PAINT's window
Background	Edit the background
Channel	Sets this channel as the active one.
Document	Activates the document
Frame	ACTIVATE
Frames	ACTIVATE
Layer	Makes this layer the active one
Window	Sets this window to be the active one.
Workspace	Activates the workspace

Add Method

Select one of the available subtopics below to see detailed help on **Add** method

Object	Method description
Channels	ADD
ColorMaskColors	ADD
LayerRange	ADD
Layers	ADD

AddClipMask Method

Select one of the available subtopics below to see detailed help on **AddClipMask** method

Object	Method description
LayerRange	ADDCLIPMASK
ObjectLayer	ADDCLIPMASK

AddColor Method

Select one of the available subtopics below to see detailed help on **AddColor** method

Object	Method description
ColorMask	ADDCOLOR
ColorTable	Create a new color table.
Palette	Adds a color to a custom color palette

AffineDistort Method

Select one of the available subtopics below to see detailed help on **AffineDistort** method

Object	Method description
Layer	AFFINEDISTORT
LayerRange	AFFINEDISTORT
Mask	Used by CorelSCRIPT to record manual manipulation of a mask area. If you want to distort a mask in Corel SCRIPT, use the MaskDistort command.

AlignToDocument Method

Select one of the available subtopics below to see detailed help on **AlignToDocument** method

Object	Method description
Layer	ALIGNTODOCUMENT
LayerRange	ALIGNTODOCUMENT
Mask	ALIGNTODOCUMENT

AlignToDocumentCenter Method

Select one of the available subtopics below to see detailed help on **AlignToDocumentCenter** method

Object	Method description
Layer	ALIGNTODOCUMENTCENTER
LayerRange	ALIGNTODOCUMENTCENTER
Mask	ALIGNTODOCUMENTCENTER

AlignToGrid Method

Select one of the available subtopics below to see detailed help on **AlignToGrid** method

Object	Method description
Layer	ALIGNTOGRID
LayerRange	ALIGNTOGRID
Mask	ALIGNTOGRID

AlignToLayer Method

Select one of the available subtopics below to see detailed help on **AlignToLayer** method

Object	Method description
Layer	ALIGNTOLAYER
LayerRange	ALIGNTOLAYER
Mask	ALIGNTOLAYER

AlignToLayerRange Method

Select one of the available subtopics below to see detailed help on **AlignToLayerRange** method

Object	Method description
Layer	ALIGNTOLAYERRANGE
LayerRange	ALIGNTOLAYERRANGE
Mask	ALIGNTOLAERRANGE

AlignToPoint Method

Select one of the available subtopics below to see detailed help on **AlignToPoint** method

Object	Method description
Layer	ALIGNTOPOINT
LayerRange	ALIGNTOPOINT
Mask	ALIGNTOPOINT

Clear Method

Select one of the available subtopics below to see detailed help on **Clear** method

Object	Method description
Background	Clear the background
Clipboard	Clears the contents of the clipboard

Close Method

Select one of the available subtopics below to see detailed help on **Close** method

Object	Method description
Document	Closes the document
Palette	Closes the color palette
Window	Closes the window.

Combine Method

Select one of the available subtopics below to see detailed help on **Combine** method

Object	Method description
DropShadow	Combines the dropshadow with the object.
LayerRange	COMBINE

ConvertToBW Method

Select one of the available subtopics below to see detailed help on **ConvertToBW** method

Object	Method description
Color	Converts the color model to Black and White
Document	Converts the document into the black-and-white mode

Copy Method

Select one of the available subtopics below to see detailed help on **Copy** method

Object	Method description
Background	Copy the background
Layer	COPY
LayerRange	COPY

CreateMask Method

Select one of the available subtopics below to see detailed help on **CreateMask** method

Object	Method description
Channel	Creates a new mask based on the channel.
Layer	CREATEMASK
LayerRange	CREATEMASK

CreatePalette Method

Select one of the available subtopics below to see detailed help on **CreatePalette** method

Object	Method description
Document	Creates a new palette
Layer	CREATEPALETTE
LayerRange	CREATEPALETTE

CreateSelection Method

Select one of the available subtopics below to see detailed help on **CreateSelection** method

Object	Method description
Layer	Selects this layer only
LayerRange	CREATESELECTION

CropToMask Method

Select one of the available subtopics below to see detailed help on **CropToMask** method

Object	Method description
Document	Crops the selected image to the bounding box of the current mask
Layer	CLIPTOMASK
LayerRange	CLIPTOMASK

Cut Method

Select one of the available subtopics below to see detailed help on **Cut** method

Object	Method description
Background	Cut the background
Layer	CUT
LayerRange	CUT

Defringe Method

Select one of the available subtopics below to see detailed help on **Defringe** method

Object	Method description
LayerRange	DEFRINGE
ObjectLayer	DEFRINGE

Delete Method

Select one of the available subtopics below to see detailed help on **Delete** method

Object	Method description
Channel	Deletes the channel.
ClipMask	Deletes the clipmask.
ColorMaskColor	DELETE
CorelScriptFile	Deletes the Corel SCRIPT file from disk
DropShadow	Deletes the dropshadow.
Floater	Deletes the floater.
Frame	Deletes the frame.
Frames	Deletes the frames within the specified range.
Guide	Deletes the guides
Layer	Deletes the selected layer(s).
LayerRange	DELETE
Mask	Delete Mask

Distort Method

Select one of the available subtopics below to see detailed help on **Distort** method

Object	Method description
Layer	Distorts the shape of the selected layer.
LayerRange	DISTORT
Mask	Distorts the shape of the current mask.

Duplicate Method

Select one of the available subtopics below to see detailed help on **Duplicate** method

Object	Method description
Document	Creates a copy of the image and assigns it the specified name
Layer	Duplicates the selected layer(s). The new layers are placed in front of existing layers and remain selected. The original layers are deselected.
LayerRange	DUPLICATE

Feather Method

Select one of the available subtopics below to see detailed help on **Feather** method

Object	Method description
Layer	Feathers the edges of the selected layer(s).
LayerRange	FEATHER
Mask	Feathers the edges of the mask, creating a smoothing effect.

Flip Method

Select one of the available subtopics below to see detailed help on **Flip** method

Object	Method description
Document	Flips the image horizontally and/or vertically
Layer	FLIP
LayerRange	FLIP
Mask	FLIP

GetIndexOfColor Method

Select one of the available subtopics below to see detailed help on **GetIndexOfColor** method

Object	Method description
ColorTable	Gets the index of a color in the palette.
Palette	Gets the index of a color object

GetPosition Method

Select one of the available subtopics below to see detailed help on **GetPosition** method

Object	Method description
Floater	Gets the position of the floater.
Layer	GETPOSITION
LayerRange	GETPOSITION
Mask	GETPOSITION

GetSize Method

Select one of the available subtopics below to see detailed help on **GetSize** method

Object	Method description
Layer	GETSIZE
LayerRange	GETSIZE
Mask	GETSIZE

Load Method

Select one of the available subtopics below to see detailed help on **Load** method

Object	Method description
Channels	LOAD
Mask	Load mask

Merge Method

Select one of the available subtopics below to see detailed help on **Merge** method

Object	Method description
Layer	Merges the selected layer(s) or all layers with the background. The layers(s) become(s) part of the image and cannot be selected again.
LayerRange	MERGE
Layers	MERGE

Move Method

Select one of the available subtopics below to see detailed help on **Move** method

Object	Method description
Floater	Moves the floater by a given delta.
Frame	Move the frame to the given index.
Frames	Moves the specified frame to a given index.
Layer	MOVE
LayerRange	MOVE
Mask	MOVE

OrderBackOf Method

Select one of the available subtopics below to see detailed help on **OrderBackOf** method

Object	Method description
Layer	ORDERBACKOF
LayerRange	ORDERBACKOF

OrderBackOne Method

Select one of the available subtopics below to see detailed help on **OrderBackOne** method

Object	Method description
Layer	ORDERBACKONE
LayerRange	ORDERBACKONE

OrderForwardOne Method

Select one of the available subtopics below to see detailed help on **OrderForwardOne** method

Object	Method description
Layer	ORDERFORWARDONE
LayerRange	ORDERFORWARDONE

OrderFrontOf Method

Select one of the available subtopics below to see detailed help on **OrderFrontOf** method

Object	Method description
Layer	ORDERFRONTOF
LayerRange	ORDERFORWARDOF

OrderToBack Method

Select one of the available subtopics below to see detailed help on **OrderToBack** method

Object	Method description
Layer	ORDERTOBACK
LayerRange	ORDERTOBACK

OrderToFront Method

Select one of the available subtopics below to see detailed help on **OrderToFront** method

Object	Method description
Layer	ORDERTOFRONT
LayerRange	ORDERTOFRONT

RemoveMatte Method

Select one of the available subtopics below to see detailed help on **RemoveMatte** method

Object	Method description
LayerRange	REMOCEMATTE
ObjectLayer	REMOVEMATTE

Rotate Method

Select one of the available subtopics below to see detailed help on **Rotate** method

Object	Method description
Document	Rotates the image to a specified degree
Layer	Rotates the selected layer(s) by the specified angle, about the given rotation point.
LayerRange	ROTATE
Mask	ROTATE

Save Method

Select one of the available subtopics below to see detailed help on **Save** method

Object	Method description
Document	Saves the document using its current name and location
Palette	Saves the palette

SetPosition Method

Select one of the available subtopics below to see detailed help on **SetPosition** method

Object	Method description
Floater	Sets the position of the floater.
Layer	SETPOSITION
LayerRange	SETPOSITION
Mask	SETPOSITION

SetSize Method

Select one of the available subtopics below to see detailed help on **SetSize** method

Object	Method description
Layer	SETSIZE
LayerRange	SETSIZE
Mask	SETSIZE

Skew Method

Select one of the available subtopics below to see detailed help on **Skew** method

Object	Method description
Layer	Skews the selected layer(s).
LayerRange	SKEW
Mask	Skews the active mask.

Split Method

Select one of the available subtopics below to see detailed help on **Split** method

Object	Method description
Document	Separates an image into the color channels corresponding to the specified color model
DropShadow	Creates a new object with the dropshadow.

Stretch Method

Select one of the available subtopics below to see detailed help on **Stretch** method

Object	Method description
Layer	Stretches the selected layer(s).
LayerRange	STRETCH
Mask	Stretches the active mask.

Threshold Method

Select one of the available subtopics below to see detailed help on **Threshold** method

Object	Method description
Layer	Alters the edges of mask selections or objects by removing the smooth transition between a selection, or a layer, and the underlying image.
LayerRange	THRESHOLD
Mask	Converts grayscale to a black and white (or bi-level) mask using the specified threshold value.

UngroupAll Method

Select one of the available subtopics below to see detailed help on **UngroupAll** method

Object	Method description
Layer	UNGROUPALL
LayerRange	UNGROUPALL

ANGLECONVERT function

ANGLECONVERT(x, y, z)

Converts a number from one angle measurement to another.

Parameter	Description
x	Any number from 1 to 5 that indicates the unit of measurement from which to convert. 1 = degrees 2 = radians 3 = gradients 4 = Corel PHOTO-PAINT degrees (tenths of a degree) 5 = CorelDRAW degrees (millionths of a degree)
y	Any number from 1 to 5 that indicates the unit of measurement to convert to. 1 = degrees 2 = radians 3 = gradients 4 = Corel PHOTO-PAINT degrees (tenths of a degree) 5 = CorelDRAW degrees (millionths of a degree)
z	Any numeric expression specifying the value to be converted.

Example

x_rads = ANGLECONVERT(1, 2, 90)

The above example converts 90 degrees to radians. The variable **x_rads** equals 1.57142857142932.

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

FROMCENTIMETERS function

FROMCENTIMETERS(x)

Converts a numeric value from centimeters to tenths of a micron.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.CreateRectangle FROMCENTIMETERS(8), FROMCENTIMETERS(-5), FROMCENTIMETERS(0), FROMCENTIMETERS(2.5), FROMCENTIMETERS(0.75)

This CorelDRAW command would create a rectangle 7.5 by 8 centimeters. The rectangle's top left corner coordinate is -5, 8 centimeters relative to the center of the page, and the corners are 0.75 centimeters in diameter.

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

FROMCICEROS function

FROMCICEROS(x)

Converts a numeric value from ciceros to tenths of a micron.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.CreateRectangle FROMCICEROS(18), FROMCICEROS(-12), FROMCICEROS(8), FROMCICEROS(6), FROMCICEROS(1.5)

This CorelDRAW command would create a rectangle 18 by 10 ciceros. The rectangle's top left corner coordinate is -12, 18 ciceros relative to the center of the page, and the corners are 1.5 ciceros in diameter.

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

FROMDIDOTS function

FROMDIDOTS(x)

Converts a numeric value from didots to tenths of a micron.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.CreateRectangle FROMDIDOTS(50), FROMDIDOTS(-70), FROMDIDOTS(0), FROMDIDOTS(30), FROMDIDOTS(20)

This CorelDRAW command would create a rectangle 100 by 50 didots. The rectangle's top left coordinate is -70, 50 didots relative to the center of the page, and the corners are 20 didots in diameter.

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

FROMINCHES function

FROMINCHES(x)

Converts a numeric value from inches to tenths of a micron.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.CreateRectangle FROMINCHES(3), FROMINCHES(-2), FROMINCHES(0), FROMINCHES(1), FROMINCHES(0.25)

This CorelDRAW command would create a rectangle 3 by 3 inches. The rectangle's top left coordinate is -2, 3 inches relative to the center of the page, and the corners are 0.25 inches in diameter.

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

FROMPICAS function

FROMPICAS(x)

Converts a numeric value from picas to tenths of a micron.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.CreateRectangle FROMPICAS(18), FROMPICAS(-12), FROMPICAS(8), FROMPICAS(6), FROMPICAS(2)

This CorelDRAW command would create a rectangle 18 by 10 picas. The rectangle's top left coordinate is -12, 18 picas relative to the center of the page, and the corners are 2 picas in diameter.

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

FROMPOINTS function

FROMPOINTS(x)

Converts a numeric value from points to tenths of a micron.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.CreateRectangle FROMPOINTS(210), FROMPOINTS(-140), FROMPOINTS(90), FROMPOINTS(70), FROMPOINTS(1.75)

This CorelDRAW command would create a rectangle 210 by 140 points. The rectangle's top left corner coordinate is -140, 210 points relative to the center of the page, and the corners are 1.75 points in diameter.

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

LENGTHCONVERT function

LENGTHCONVERT(x, y, z)

Converts a number from one length measurement to another.

Parameter	Description
x	Any number from 1 to 7 that indicates the unit of measurement from which to convert. 1 inches 2 centimeters 3 points 4 Ciceros 5 didots 6 picas 7 CorelDRAW and VENTURA units (tenths of a micron)
y	Any number from 1 to 7 that indicates the unit of measurement to convert to: 1 inches 2 centimeters 3 points 4 Ciceros 5 didots 6 picas 7 CorelDRAW and VENTURA units (tenths of a micron)
z	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

x_microns = LENGTHCONVERT(1, 7, 1)

The above example converts one inch to tenths of a micron. The variable **x_microns** equals 254,000.

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

TOCENTIMETERS function

TOCENTIMETERS(x)

Converts a numeric value from tenths of a micron to centimeters.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.GetPosition (x,y)

xCM = TOCENTIMETERS (x)

yCM = TOCENTIMETERS (y)

In this CorelDRAW example, **xCM** and **yCM** are set to the X and Y coordinates of the selected object in centimeters.

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

TOCICEROS function

TOCICEROS(x)

Converts a numeric value from tenths of a micron to ciceros.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.GetPosition (x,y)

xCiceros = TOCICEROS (x)

yCiceros = TOCICEROS (y)

In this CorelDRAW example, **xCiceros** and **yCiceros** are set to the X and Y coordinates of the selected object in ciceros.

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

TODIDOTS function

TODIDOTS(x)

Converts a numeric value from tenths of a micron to didots.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.GetPosition (x,y)

xDidots = TODIDOTS (x)

yDidots = TODIDOTS (y)

In this CorelDRAW example, **xDidots** and **yDidots** are set to the X and Y coordinates of the selected object in didots.

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

TOINCHES function

TOINCHES(x)

Converts a numeric value from tenths of a micron to inches.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.GetPosition (x,y)

xInch = TOINCHES (x)

yInch = TOINCHES (y)

In this CorelDRAW example, **xInch** and **yInch** are set to the X and Y coordinates of the selected object in inches.

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

TOPICAS function

TOPICAS(x)

Converts a numeric value from tenths of a micron to picas.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.GetPosition (x,y)

xPica = TOPICAS (x)

yPica = TOPICAS (y)

In this CorelDRAW example, **xPica** and **yPica** are set to the X and Y coordinates of the selected object in picas.

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

TOPOINTS function

TOPOINTS(x)

Converts a numeric value from tenths of a micron to points.

Parameter	Description
x	Any numeric expression specifying the value to be converted.

Note

- Tenths of a micron are used as the basic unit of measurement in Corel applications such as CorelDRAW and Corel VENTURA.

Example

.GetPosition (x,y)

xPoint = TOPOINTS (x)

yPoint = TOPOINTS (y)

In this CorelDRAW example, **xPoint** and **yPoint** are set to the X and Y coordinates of the selected object in points.

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

GETCOLOR statement and function

Statement syntax
GETCOLOR Red, Green, Blue

Function syntax
ReturnValue = GETCOLOR (Red, Green, Blue)

Displays a standard Windows Color dialog box and returns color setting values from the RGB color model (Red, Green, Blue).

Return Value

The GETCOLOR function returns one of the following values

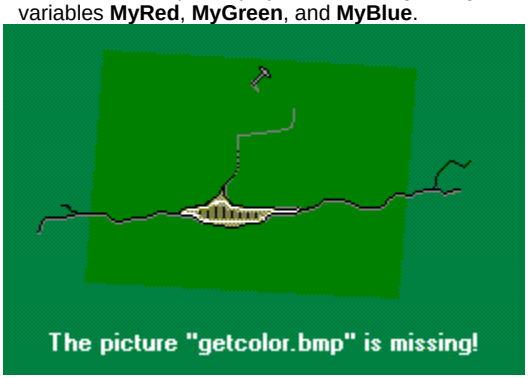
- TRUE (-1) — the Cancel button was not clicked
- FALSE (0) — the Cancel button was clicked

Parameter	Definition
Red	Lets you specify the numeric variable that is passed the Red setting of the selected color (0 - 255). You can also use this variable to set an initial value.
Green	Lets you specify the numeric variable that is passed the Green setting of the selected color (0 - 255). You can also use this variable to set an initial value.
Blue	Lets you specify the numeric variable that is passed the Blue setting of the selected color (0 - 255). You can also use this variable to set an initial value.

Example

GETCOLOR MyRed%, MyGreen%, MyBlue%

The above example displays the following dialog box and returns the RGB color settings for the selected color to the numeric variables **MyRed**, **MyGreen**, and **MyBlue**.



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

GETFILEBOX function

GETFILEBOX(Filter, Title, Type, DefFile, DefExt, DefFol, BtnName**)**

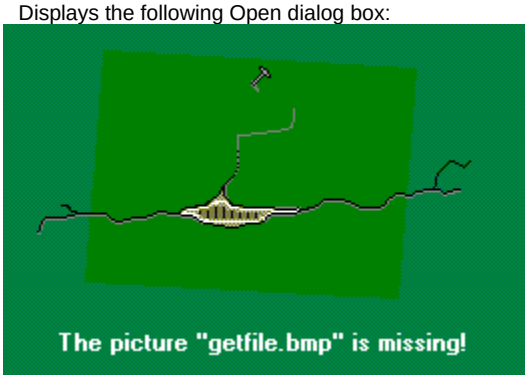
This function displays a standard Windows File Open or File Save As dialog box. Both dialog boxes allow users to choose a file from the file system. The **GETFILEBOX** function returns the selected filename and its full path, or an empty string if the user chooses Cancel. The **GETFILEBOX** statement by itself does not open or save a file; it only returns a string corresponding to the selected file.

Parameter	Definition
Filter	String expression specifying the filters to use in the dialog box. For the Open dialog box, the filters are listed in the Files of Type list box. For the Save As dialog box, the filters are listed in the Save as Type list box. Filters are specified in two parts. The first part is the text that appears in the list box, and the second part is the actual filter extension. The parts are separated by the | character (do not use spaces before or after the | characters). To separate multiple filters, use the | character. See the example below for more information.
Title	String expression specifying the title to display in the dialog box. If not specified, "Open" is displayed for an Open dialog box and "Save As" is displayed for a Save As dialog box.
Type	Numeric expression specifying the type of dialog box to display: 0 File Open dialog box (default if omitted) 1 File Save dialog box
DefFile	String expression specifying the text to display in the File name text box of the dialog box. If not specified, the text box is empty.
DefExt	String expression specifying the default extension to append to a File name if the user omits the extension.
DefFol	String expression specifying the default folder used by the dialog box. If not specified, or the specified folder does not exist, the current folder is used.
BtnName	String expression specifying a button name to override the Open or Save button in the dialog box. If not specified, the button's name remains unchanged.

Example

```
SETCURRFOLDER = "c:\COREL50\DRAW\samples" 'set the current folder
Filename$=GETFILEBOX("Included Scripts|*.csc|All Files|*.*", "Scripts included...", 0,"animals")
```

Displays the following Open dialog box:



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

GETFOLDER function

GETFOLDER (InitFolder**)**

This function displays a Windows Choose Folder dialog box. The Choose Folder dialog box returns the folder and path a user chooses as a string.

Parameter	Definition
InitFolder	String expression specifying the default path and folder to display in the dialog box. If not specified, the active folder is used.

Note

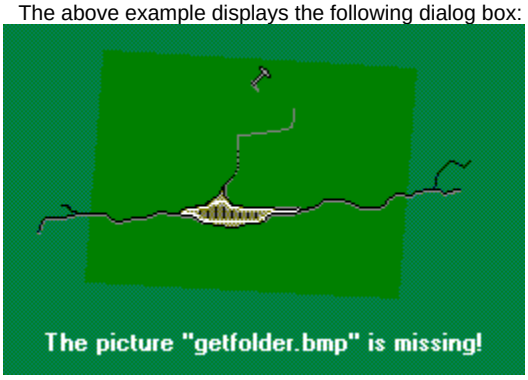
- If the Cancel button is clicked, an empty string is returned.

Example

NewFolder$ = GETFOLDER("D:\Corel60")

SETCURRFOLDER = NewFolder$

The above example displays the following dialog box:



The selected folder is passed to the string variable **NewFolder**. The **SETCURRFOLDER** statement sets the current folder to the folder name passed to **NewFolder**.

{button ,AL(`cs_ui_statements;chfolder;currfolder;;;',0,"Defaultoverview",)} Related Topics

GETFONT statement and function

Statement syntax
GETFONT FaceName, PointSize, Weight, Italic, Underline, StrikeOut, Red, Green, Blue

Function syntax
ReturnValue = GETFONT (FaceName, PointSize, Weight, Italic, Underline, StrikeOut, Red, Green, Blue)

Displays a standard Windows Font dialog box and returns the selected font settings.

Return Value

The GET font function returns one of the following values:

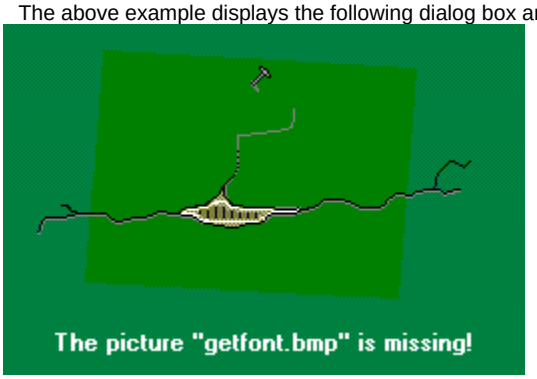
- TRUE (-1) the Cancel button was not clicked
- FALSE (0) the Cancel button was clicked

Parameter	Definition
FaceName	Lets you specify a string variable that is passed the name of the selected font. You can also use this variable to set an initial value.
PointSize	Lets you specify a numeric variable that is passed the font size in points. You can also use this variable to set an initial value. This parameter uses non-fractional values
Weight	Lets you specify a numeric variable that is passed the font's weight setting (number of inked pixels per 1000 pixels). Common values and their corresponding names include: 100 Thin 200 Extra Light, Ultra Light 300 Light **400 Normal, Regular** 500 Medium 600 Semi Bold, Demi Bold **700 Bold** 800 Extra Bold, Ultra Bold 900 Black, Heavy Most Windows fonts only use two weight settings: 400 (Normal) and 700 (Bold).
Italic	Lets you specify a numeric variable that is passed the font's italic setting: TRUE (-1) if this setting is enabled; FALSE (0) otherwise.
Underline	Lets you specify a numeric variable that is passed the font's underline setting: TRUE (-1) if this setting is enabled; FALSE (0) otherwise.
StrikeOut	Lets you specify a numeric variable that is passed the font's strike out setting: TRUE (-1) if this setting is enabled; FALSE (0) otherwise.
Red	Lets you specify the numeric variable that is passed the Red (RGB color model) setting of the selected font's color (0 - 255). You can also use this variable to set an initial value.
Green	Lets you specify the numeric variable that is passed the Green (RGB color model) setting of the selected font's color (0 - 255). You can also use this variable to set an initial value.
Blue	Lets you specify the numeric variable that is passed the Blue (RGB color model) setting of the selected font's color (0 - 255). You can also use this variable to set an initial value.

Example

GETFONT FN, PS, Wt, Italic, UL, SO, R, G, B

The above example displays the following dialog box and returns the font settings the variables specified.



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

INPUTBOX function

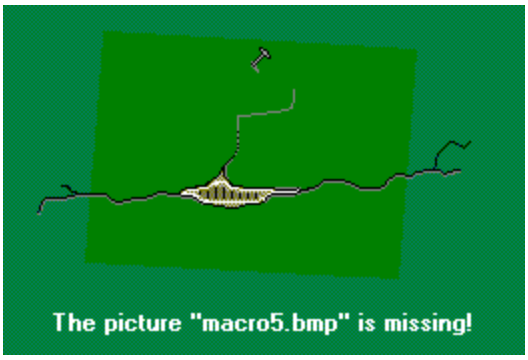
INPUTBOX(prompt$)

Displays a simple dialog box where you can enter a string that is returned to a script. The dialog box has OK and Cancel buttons. If the Cancel button is chosen, an empty string is returned.

Parameter	Definition
prompt$	String expression that appears in the dialog box above the edit box.

Example

MyString$ = INPUTBOX("Please type in a string")



User input is returned to the variable **MyString$.**

{button ,AL(`cs_ui_statements;textbox;text;;;',0,"Defaultoverview",)} Related Topics

MESSAGE statement

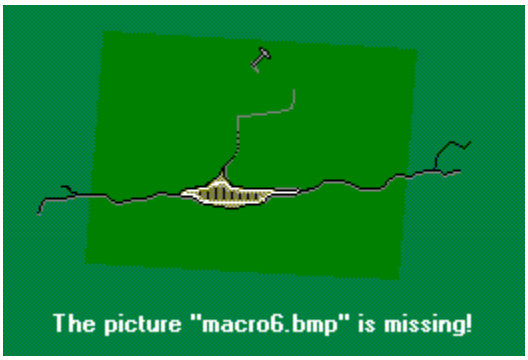
Message anyVariable

Displays a dialog box that contains a specified message and an OK button. This statement doesn't return any value to a running script, but can provide the script user with information during script execution.

Parameter	Definition
anyVariable	Any numeric or string expression to display in the message box. Numbers and dates are displayed as their string representations.

Example

```
x$="Hello." + CHR(13)     'CHR(13) is a return character
MESSAGE x$ + "What a nice day."
```



Note

- See the Corel SCRIPT character map for a list of character codes.

{button ,AL(`cs_ui_statements;textbox;text;chr;;',0,"Defaultoverview",)} Related Topics

MESSAGEBOX function

MESSAGEBOX(prompt, title, option**)**

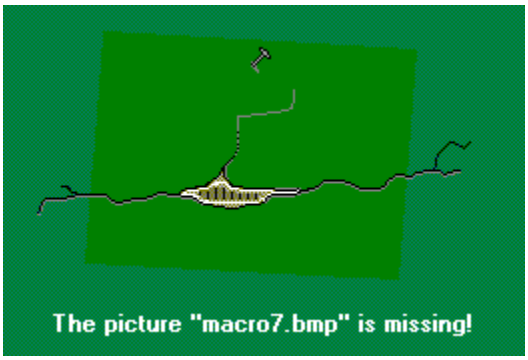
Displays a message box with a specified message and user-specified buttons and icons. MESSAGEBOX returns a value representing the button that was used to close it to the script.

Parameter	Definition
prompt	String expression to display in the box.
title	String expression to display in the message box caption.
option	A numeric expression representing the type of buttons to include in the box and an icon (if any) to appear beside the message. The **option** value is set using the OR (or +) operator for multiple buttons (see example): **Button Type:** 0 OK only (used by default if a button is not specified) 1 OK/Cancel 2 Abort/Retry/Ignore 3 Yes/No/Cancel 4 Yes/No 5 Retry/Cancel **Icon Type** (click hot spot for an example)**:** 0 No icon 16 Stop 32 Question 48 Exclamation 64 Information

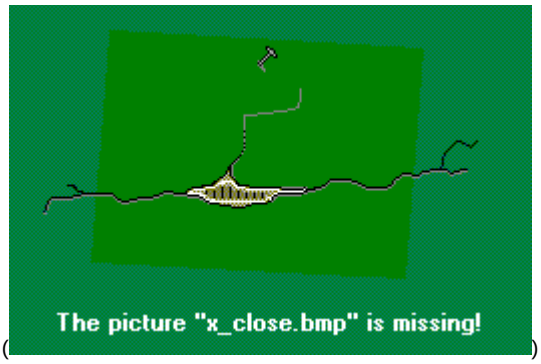
Returns	Button pressed
1	OK
2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

Example

retval% = MESSAGEBOX("This dialog displays three buttons", "MESSAGEBOX Example", 3 OR 48)

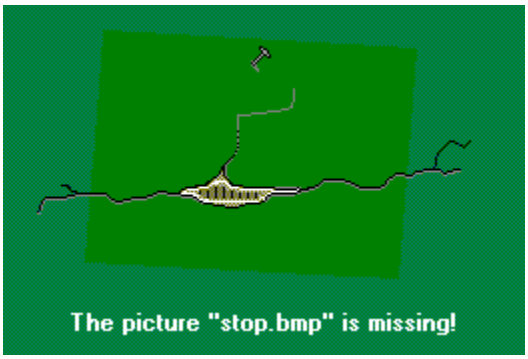


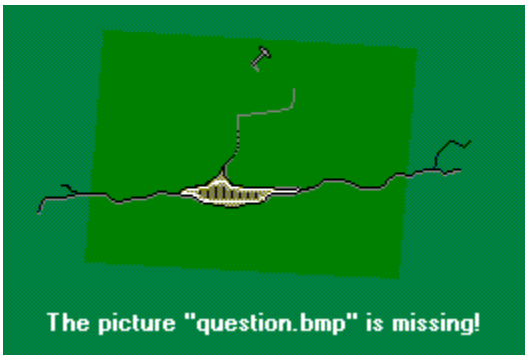
Note

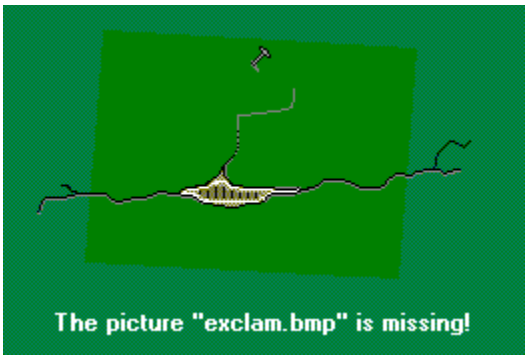
- Pressing the Close Dialog button (

) is the same as pressing the Cancel button; both return 2.

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


The picture "stop.bmp" is missing!


The picture "question.bmp" is missing!


The picture "exclam.bmp" is missing!


The picture "info.bmp" is missing!

BUILDDATE function

BUILDDATE (Year, Month, Day)

Assigns a date value to a date variable.

Parameter	Description
Year	Numeric expression specifying the year to assign to a date variable.
Month	Numeric expression specifying the month to assign to a date variable. Valid values are from 1 to 12 inclusive.
Day	Numeric expression specifying the day to assign to a date variable. Valid values are from 1 to 31 inclusive, depending on the **Month** setting.

Note

- **BUILDATE** can only accept a date value between January 1, 1980 (date serial number 29221) and December 31, 2099 (date serial number 73050). If a date outside this range is specified, an error occurs.

Example

```
DIM BigDay AS DATE
BigDay = BUILDDATE(1996, 1, 14)
```

In the above example, the first line declares the date variable **BigDay**. This variable is then assigned the date January 14, 1996.

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

BUILDTIME function

BUILDTIME (Hour, Minute, Second)

Assigns a time value to a date variable.

Parameter	Description
Hour	Numeric expression specifying the hour to assign to a date variable. This value is based on a 24-hour clock. For example, 5 PM equals 17. Valid values are from 0 to 23 inclusive.
Minute	Numeric expression specifying the minute to assign to a date variable. Valid values are from 0 to 59 inclusive.
Second	Numeric expression specifying the second to assign to a date variable. Valid values are from 0 to 59 inclusive.

Example

```
DIM BigDay AS DATE
BigDay = BUILDTIME(14, 30, 0)
```

In the above example, the first line declares the date variable **BigDay**. This variable is then assigned the time 2:30 PM.

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

FORMATDATE function

DateString = FORMATDATE (DateExp, FormatString)

Converts a date expression into a string with a specified format.

Return Value

Lets you specify the string variable that is assigned the formatted date.

Parameter	Description
DateExp	Lets you specify the date expression to convert to a string.
FormatString	Lets you specify a code, as a string, representing the format of the date. Formats are created by using and combining the following codes.

To format	Use this format code (case-sensitive)
Days as 1-31	d
Days as 01-31	dd
Days as Sun-Sat	ddd
Days as Sunday-Saturday	dddd
Months as 1-12	M
Months as 01-12	MM
Months as Jan-Dec	MMM
Months as January-December	MMMM
Year as 6 in 1996	y
Year as 96 in 1996	yy
Year as 1996	yyy

Note

- You can insert spaces and punctuation between date elements within the formatting string. See the example below.

Example

```
DIM TodayDate AS DATE
TodayDate = GETCURRDATE()
StringDate$ = FORMATDATE (TodayDate, "dddd, MMMM d, yyy")
MESSAGE StringDate
```

In the above example, the first line declares the date variable **TodayDate**. This variable is then assigned the current date with the **GETCURRDATE** function. The **StringDate** variable is then assigned today's date using the formatting shown in the following example.

Saturday, September 16, 1995

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

FORMATTIME function

TimeString = FORMATTIME (TimeExp, FormatString)

Converts a time expression into a string with a specified format.

Return Value

Lets you specify the string variable that is assigned the formatted time.

Parameter	Description
TimeExp	Lets you specify the time expression to convert to a string.
FormatString	Lets you specify a code, as a string, representing the format of the time. Formats are created by using and combining the following codes.

To format	Use this format code (case-sensitive)
Hours as 1-12 (12-hour clock)	h
Hours as 01-12 (12-hour clock)	hh
Hours as 0-23 (24-hour clock)	H
Hours as 00-23 (24-hour clock)	HH
Minutes as 0-59	m
Minutes as 00-59	mm
Seconds as 0-59	s
Seconds as 00-59	ss
AM/PM as A or P	t
AM/PM as AM or PM	tt
Time as 4:36 pm	h:mm pm

Note

- You can insert spaces and punctuation between time elements within the formatting string. See the example below.

Example

```
DIM TimeNow AS DATE
TimeNow = GETCURRDATE()
StringTime = FORMATTIME (TimeNow, "HH:mm:ss tt")
MESSAGE StringTime
```

In the above example, the first line declares the date variable **TodayDate**. This variable is then assigned the current date and time with the **GETCURRDATE** function. The **StringTime** variable is then assigned the current time using the formatting shown in the following example.

```
13:46:25 PM
```

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

GETCURRDATE function

DateTime = GETCURRDATE ( )

Returns the system's current date and time.

Return Value

Lets you specify the date variable that is assigned the current date and time. See Assigning values to date variables for more information.

Note

- Formatting used to display dates and time is set in the Windows Control Panel. In Windows 95, see Regional settings for formatting information; in Windows NT, see International settings.
- **GETCURRDATE** can return a date between January 1, 1980 (date serial number 29221) and December 31, 2099 (date serial number 73050).
- In Corel SCRIPT version 7.0, the **GETCURRDATE** function and the **SETCURRDATE** statement replace the **CURRDATE** statement.

Example

```
DIM DT AS DATE
DT = GETCURRDATE()
MESSAGE DT
```

The above example displays the system date and time in a message box. You can also extract portions of the date and convert them to strings and numbers. The following continues the above example:

```
MyDateString$ = DT                ' create a string
MyDay% = VAL(LEFT(MyDateString$, 2)) ' extract the day as an integer
```

In the above example, the **VAL** and **LEFT** functions are used to extract an integer from the **MyDateString** string variable (created by converting a date variable). The method you use to extract portions from a date string depend on your Windows date settings. You can also use the **GETDATEINFO** or the **GETTIMEINFO** function to extract information from a date variable.

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

GETDATEINFO function

GETDATEINFO DateExp, Year, Month, Day, DayOfWeek

Extracts the components of a date expression to numeric variables.

Parameter	Description
DateExp	Lets you specify the date expression to extract components from.
Year	Lets you specify the numeric variable that is assigned the year component from the specified date expression.
Month	Lets you specify the numeric variable that is assigned the month component from the specified date expression.
Day	Lets you specify the numeric variable that is assigned the day component from the specified date expression.
DayOfWeek	Lets you specify the numeric variable that is assigned the day of week component from the specified date expression. Sunday corresponds to 1, Monday to 2, and so on.

Note

- **GETDATEINFO** can only accept a date value between January 1, 1980 (date serial number 29221) and December 31, 2099 (date serial number 73050). If a date outside this range is specified, an error occurs.

Example

```
DIM TodayDate AS DATE

TodayDate = GETCURRDATE()

GETDATEINFO TodayDate, Y&, M&, D&, DW&
```

In the above example, the first line declares the date variable **TodayDate**. This variable is then assigned the current date with the **GETCURRDATE** function. The variables **Y**, **M**, **D**, and **DW** are then assigned their respective component of the date stored in **TodayDate**. If **TodayDate** was set to May 29, 1996, then **Y**=1996, **M**=5, **D**=29, and **DW**=4.

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

GETTIMEINFO function

GETTIMEINFO TimeExp, Hour, Minute, Second

Extracts the components of a time expression to numeric variables.

Parameter	Description
TimeExp	Lets you specify the time expression to extract components from.
Hour	Lets you specify the numeric variable that is assigned the hour component from the specified time expression. The number assigned is based on a 24-hour clock. For example, 16 is the numeric variable for 4pm.
Minute	Lets you specify the numeric variable that is assigned the minute component from the specified time expression.
Second	Lets you specify the numeric variable that is assigned the second component from the specified time expression.

Example

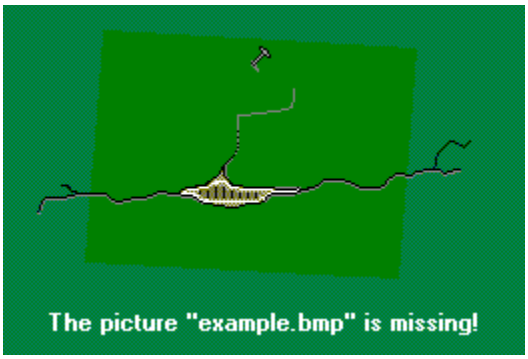
```
DIM TodayTime AS DATE

TodayTime = GETCURRDATE()

GETTIMEINFO TodayTime, H&, M&, S&
```

In the above example, the first line declares the date variable **TodayTime**. This variable is then assigned the current date and time with the **GETCURRDATE** function. The variables **H**, **M**, and **S** are then assigned their respective component of the time stored in **TodayTime**. If **TodayTime** was set to 5:37:16 PM, then **H**=17, **M**=37, **S**=16.

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



SETCURRDATE statement

SETCURRDATE DT

Sets the system's date and time. If used improperly, this statement can cause problems in the system's Windows settings.

Parameter	Description
DT	Lets you specify a date expression that sets the system's date and time

Note

- **SETCURRDATE** can assign a date between January 1, 1980 (date serial number 29221) and December 31, 2099 (date serial number 73050).
- Formatting used to display dates and time is set in the Windows Control Panel. In Windows 95, see Regional settings for formatting information; in Windows NT, see International settings.
- In Corel SCRIPT version 7.0, the **SETCURRDATE** statement and the **GETCURRDATE** function replace the **CURRDATE** statement.

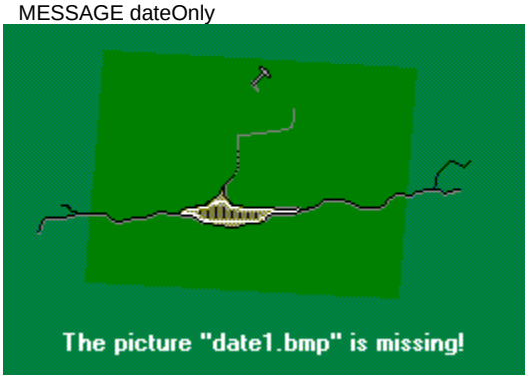
{button ,AL(`cs_date_time;;;;;',0,"Defaultoverview",)} Related Topics

Examples for SETCURRDATE

To change the system date and time

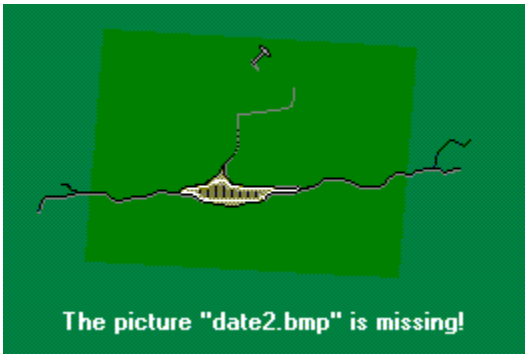
The following example sets the system date to August 25, 1995, and the time to 12:00 A.M. A message box is used to display the date value of **dateOnly**.

```
DIM dateOnly AS DATE
dateOnly = 34936
SETCURRDATE dateOnly
MESSAGE dateOnly
```



The following example sets the system date to August 25, 1995, and the system time to 12:05.46 P.M. A message box is used to display the date value of **dateAndTime.**

```
DIM dateAndTime AS DATE
dateAndTime = 34936.504
CURRDATE = dateAndTime
MESSAGE dateAndTime
```



Using SETCURRDATE for file date stamping

The following example assigns the system date to the variable **xDate**. The third line sets the system date to February 26, 1994, and the fourth line opens a text file that is stamped with the new system date. The last line resets the system date to its original value.

```
DIM xDate AS DATE
xDate = GETCURRDATE      'assigns system date
SETCURRDATE "02/26/1994"     'sets the system date
OPEN "c:\log.txt" FOR OUTPUT AS 2
SETCURRDATE xDate
```

Note

- Formatting used to display dates and time is set in the Windows Control Panel. In Windows 95, see Regional settings for formatting information; in Windows NT, see International settings.

{button ,AL(`mid;left;val;cs_date_time;;',0,"Defaultoverview",)} Related Topics

WAIT FOR statement

WAIT FOR x

Pauses script execution for a specified number of seconds.

Argument	Description
x	A non-negative numeric expression specifying the number of seconds to pause script execution.

Example

MESSAGE "Start"

WAIT FOR 3

MESSAGE "Done"

The above example displays a message box, once the message box is closed, script execution pauses for three seconds, and then displays another message box.

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

WAIT UNTIL statement

WAIT UNTIL x

Pauses script execution until the system timer matches a specified date serial number.

You should consider using the **WAIT UNTIL** statement to run scripts in off-peak times. For example, if you have a large print job, you could use a script to run it during the night.

Argument	Description
x	Lets you specify the date and time when to resume execution. This parameter must be positive and greater than the system's current date.

Note

- **WAIT UNITL** can be set to a date between January 1, 1980 (date serial number 29221) and December 31, 2099 (date serial number 73050). If a date outside this range is specified, an error occurs.

Example

The following example pauses script execution until the system time matches 34936.25 (August 25, 1996, 6:00 A.M.).

```
DIM offPeak AS DATE
offPeak = 34936.25
WAIT UNTIL offPeak
```

The following example pauses script execution until 3:00 A.M. regardless of the date:

```
DIM today, tonite   AS DATE
DIM daypart AS LONG
today = CURRDATE
daypart = INT(today)   'daypart is set to 12:00 A.M. today
tonite = daypart + 1.125 'sets the date and time to 3 AM the next day
WAIT UNTIL tonite
```

If you add a day and 3 hours (1.125) to **datepart**, **tonite** is set to 3 A.M. the next morning.

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

ADDFOL statement

#ADDFOL folder

This statement adds a temporary folder to the paths Corel SCRIPT searches when trying to find an **INCLUDE** file on your system.

Parameter	Description
folder	String expression specifying a drive and a folder.

Note

- The pound sign (#) is required in the syntax.
- Corel SCRIPT searches for an INCLUDE file in the following order:

1. The folder where the script resides. You can use the **GETSCRIPTFOLDER** statement to set or determine the active folder.
2. Folders in the path. The path is specified in the systems AUTOEXEC.BAT file.
3. Folders set in the Corel SCRIPT Editors **INCLUDE** option. Click  for more information about setting INCLUDE folders.
4. Folders specified with the ADDFOL statement.

Example

```
#ADDFOL "C:\MyFiles"
```

The above example add the MYFILES folder on the C drive to the path Corel SCRIPT searches for INCLUDE files.

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

ADDRESBMP statement

#ADDRESBMP name file

This statement embeds a Windows bitmap graphic (.BMP) into an executable (.EXE), DLL, or Corel Add-on (.CAO) created with Corel SCRIPT or a Corel SCRIPT Binary file (.CSB). If you're distributing a script that uses many bitmaps in the form of an executable, DLL, or Corel SCRIPT Binary file, this statement can reduce the number of files that you must distribute.

In Corel SCRIPT, bitmaps are most often used in custom dialog boxes. See the Image dialog box control for an example of a script using bitmaps.

Parameter	Description
name	String expression specifying the bitmap reference name within a script. When this name is referenced within a script, it must be preceded with the pound sign (#) and be enclosed in quotations. However, within the **ADDRESBMP** statement, quotations should not be used. See the example below for more information.
file	String expression specifying the filename and path of a Windows bitmap graphic.

Note

- The pound sign (#) is required in the syntax.

Example

```
#ADDRESBMP Wizard1Portrait "C:\MyScripts\portrait.bmp"
BEGIN DIALOG Dialog1 200, 100, "Corel SCRIPT Dialog"
    IMAGE  11, 17, 74, 65, "#Wizard1Portrait"
END DIALOG
```

In the above example, the PORTRAIT.BMP bitmap file takes on the reference name Wizard1Portrait. This bitmap is then used in an image control in a custom dialog box. If this script was compiled into an executable the PORTRAIT bitmap would be embedded into the executable file.

{button ,AL(`image;image_dyn;Creating Corel SCRIPT Executables;Distributing_cs_exe;;',0,"Defaultoverview",)}
Related Topics

BEEP statement

BEEP

Sounds a tone.

Note

- The sound your computer makes depends on your computer's hardware (for example, sound cards, PC speaker, and so on) and the Default sound in your Windows sound settings (see your Windows Control Panel for more details).

Example

```
IF (abc<=15.3) then BEEP ELSE MESSAGE "It's greater than 15.3"
```

If the variable **abc** is less than or equal to 15.3, the computer sounds a tone.

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

BEGINWAITCURSOR and ENDWAITCURSOR statements

BEGINWAITCURSOR
ENDWAITCURSOR

The **BEGINWAITCURSOR** statement sets the mouse pointer to Busy. The Busy pointer usually appears as an hour-glass on most systems. This command is useful for scripts that perform long operations. By setting the pointer to Busy, the user is alerted that the script is still executing.

The **ENDWAITCURSOR** statement returns the pointer to its normal state. If the **ENDWAITCURSOR** statement is not in a script that uses the **BEGINWAITCURSOR** statement, the pointer reverts to its normal state after the script finishes executing.

Note

- It is good programming practice to use an **ENDWAITCURSOR** statement with every **BEGINWAITCURSOR** statement.

Example

```
BEGINWAITCURSOR
WAIT FOR 30
ENDWAITCURSOR
```

In the above example, the pointer is set to Busy, script execution is paused for 30 seconds, and then the pointer reverts to its normal state.

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

COPY statement and function

COPY file1, file2, overwrite

The **COPY** statement copies a file.

You can also use **COPY** as a function: it returns TRUE (-1) if the **COPY** operation is successful; **FALSE** (0) otherwise.

Parameter	Description
file1	String expression specifying the file to copy. **file1** can include the drive and folder.
file2	String expression specifying the where **file1** is to be copied. **file2** can include the drive and folder.
overwrite	If **file2** already exists, this determines whether to overwrite the existing file: 0 copy and overwrite (default if omitted) 1 overwrite fails

Example

```
DIM x AS STRING
DIM y AS STRING
x = "C:\work\example1.cdr"
y = "D:\work\example1.cdr"
COPY x, y, 0
```

The above example copies the EXAMPLE1.CDR file to the work folder on the D drive.

```
DIM x AS STRING
DIM y AS STRING
x = "C:\work\example1.cdr"
y = "D:\work\example1.cdr"
success = COPY (x, y, 0)
```

The above example copies the EXAMPLE1.CDR file to the Work folder on the D drive, and assigns -1 to **success**.

{button ,AL(`FINDFIRSTFOLDER_FINDNEXTFOLDER;rename;copy;SETCURRFOLDER;rmfolder;GETcurrfolder;;',0,"Defaultoverview",)} Related Topics

FILEATTR function

retval = FILEATTR(FileName)

This functions returns a file's or folder's attributes.

Return Value

The FILEATTR function returns one of the following values:

- 0 file doesn't exist
- 1 read-only files or folder
- 2 hidden files or folder
- 4 system files or folder used by operating system
- 16 folder
- 32 archive files or folder
- 128 normal files or folder
- 256 temporary files or folder
- 2048 compressed files and folders

Parameter	Description
FileName	String expression specifying the file/folder for which the attributes are returned.

Example

```
retval = FILEATTR("C:\myfiles\mysetup.txt")
```

If MYSETUP.TXT is read-only, hidden, and a system file, **retval** equals 7.

In cases where multiple attributes are returned, you can use the AND (bitwise) operator to determine specific attributes. To determine if MYSETUP.TXT was a read-only file you could use the following syntax:

```
IF 1 AND retval THEN readOnly$ = "Yes" ELSE readOnly$ = "No"
```

1 is the read-only attribute. The variable **readOnly** is assigned a string based on bitwise comparison. In this case **readOnly** is assigned "Yes".

{button ,AL(`GETCURRFOLDER;filemode;getfileattr;FINDFIRSTFOLDER_FINDNEXTFOLDER;setcurrfolder;',0,"Defaultoverview",)} Related Topics

FILEDATE function

retval = FILEDATE(FileName)

This functions returns a file's last modification date.

Return Value

Assigned the date of the specified file's last modification as date data type. If the file is not found, 0 is returned.

Parameter	Description
FileName	Lets you specify the file for which the date property is returned.

Note

- If the file is not closed when the function is executed, the function returns 12:00 AM.

Example

```
retval = FILEDATE("C:\myfiles\mytext.txt")
```

{button ,AL(`GETCURRFOLDER;filemode;getfileattr;FINDFIRSTFOLDER_FINDNEXTFOLDER;setcurrfolder;',0,"Defaultoverview",)} Related Topics

FILEMODE function

FILEMODE(num)

Returns the file mode of an open text file in your computer's memory.

Parameter	Description
num	Numeric expression specifying an open file number to examine: 1 file was opened for input 2 file was opened for output 8 file was opened for append

Note

- This the only Corel SCRIPT file function that doesn't have an optional # sign in front of the file number.

Examples

```
i% = FILEMODE(1)
```

If file 1 was opened for input, then **i%** is set to 1. If file 1 was opened for output, then **i%** is set to 2. If file 1 was opened for append, then **i%** is set to 8.

```
OPEN "C:\example.txt" FOR APPEND AS 2
i% = FILEMODE(2)
```

Assigns 8 to the variable **i%**.

{button ,AL(`OPEN_APPEND;OPEN_INPUT;OPEN_OUTPUT;;;',0,"Defaultoverview",)} Related Topics

FILEPOS function

FILEPOS (#num)

Returns the current file position of the file pointer for the specified file.

Parameter	Description
#**num**	Numeric expression specifying an open file number to examine. The # sign is optional.

Note

- The seek always starts from the first position in the file.

Example

```
OPEN "C:\HELLO.TXT" FOR INPUT AS 2
SEEK 2, 12
position% = FILEPOS(2)
```

Assigns 12 to the variable **i%**.

{button ,AL(`lof;seek;open_append;print;write;',0,"Defaultoverview",)} Related Topics

FILESIZE function

retval = FILESIZE(FileName)

Use the FILESIZE function to return a file's size in bytes.

Return Value

Assigned the size of the specified file in bytes. If the file is not found, 0 is returned.

Parameter	Description
FileName	Lets you specify the file for which the date property is returned.

Example

```
retval = FILESIZE("C:\myfiles\mytext.txt")
```

{button ,AL(`GETCURRFOLDER;filemode;getfileattr;FINDFIRSTFOLDER_FINDNEXTFOLDER;setcurrfolder;',0,"Defaultoverview",)} Related Topics

FINDFIRSTFOLDER, FINDNEXTFOLDER functions

FolderFileName$ = FINDFIRSTFOLDER(searchcriteria, attributes)
FolderFileName$ = FINDNEXTFOLDER( )

Use the **FINDFIRSTFOLDER** and **FINDNEXTFOLDER** functions to assemble or perform an operation on a list of files, folders, or both. The **FINDFIRSTFOLDER** function is used to locate the first file or first folder in a folder that meets a specified search criteria. The **FINDNEXTFOLDER** function is used to locate the next file or next folder that meets the specified search criteria set by the **FINDFIRSTFOLDER**. The **FINDNEXTFOLDER** function must be used in conjunction with the **FINDFIRSTFOLDER** function.

Return Value

String variable that is passed the name of the folder.

Parameter	Description
searchcriteria	Lets you specify the files or folders for which to search. You can include wild-card characters (* or ?).
attributes	The type of files or folders you want to use. Use the OR operator to specify multiple file and folder types: 1 read-only 2 hidden 4 system 16 Lets you specify to use folders. If not specified, files are used. 32 archive 128 normal (not read-only, hidden, system or archive file or folder) 256 temporary 2048 compressed

Note

- Specifying folder in the **attributes** parameter (16) by itself does not specify a type of folder. You must use another parameter along with 16 to specify a folder type.

Example

```
DIM DCOUNT%,  FCOUNT%             'creates 2 integer variables
DIM FILESARR$(100), DIRARR$(100)  'creates 2 string arrays

REM LOOP #1
REM FIND ALL DIRECTORIES IN THE SAMPLES FOLDER
DCOUNT = 1
DIRARR(DCOUNT) = FINDFIRSTFOLDER("D:\COREL\VENTURA\SAMPLES\*", 16 OR 128)
WHILE (DIRARR(DCOUNT) <> "")
    MESSAGE DIRARR(DCOUNT)
    IF DIRARR(DCOUNT) <> "." AND DIRARR(DCOUNT) <> ".." THEN DCOUNT = DCOUNT + 1
    DIRARR(DCOUNT) = FINDNEXTFOLDER()
WEND

REM LOOP #2
REM FIND ALL *.VP FILES IN EACH DIRECTORY FOUND IN EARLIER LOOP
DIM I%
FCOUNT = 1
FOR I% = 1 TO DCOUNT-1
    FILESARR(FCOUNT) = FINDFIRSTFOLDER("D:\COREL\VENTURA\SAMPLES\" + DIRARR(I%) + "\*.VP", 1 OR
2 OR 4 OR 32 OR 128)
    WHILE (FILESARR(FCOUNT) <> "" )
    MESSAGE DIRARR(I%) & CHR(13) & FILESARR(FCOUNT)
         FCOUNT = FCOUNT + 1
         FILESARR(FCOUNT) = FINDNEXTFOLDER()
    WEND
```

```
NEXT I%
```

In the above example, the first loop fills an array (**DIRARR**) with the names of the normal folders in the D:\COREL\VENTURA\SAMPLES folder. The second loop searches the folders in the SAMPLES folder for any file with the extension VP. Any found VP file has its name added to the **FILESARR** array and has its name displayed in a message box.

The following statement in the first loop is used to remove the current (.) folder and parent (..) folder from being sent to the **DIRARR** array:

```
IF DirArr(Dcount) <> "." AND DirArr(Dcount) <> ".." THEN Dcount = Dcount + 1
```

{button ,AL(`CURRFOLDER;filemode;getfileattr;getcurrfolder;setcurrfolder;',0,"Defaultoverview",)} Related Topics

FREEFILE function

FREEFILE ( )

Returns the lowest file number not associated with an open text file in the computer's memory.

Example

```
OPEN "temp.out" FOR OUTPUT AS 1
OPEN "temp2.out" FOR OUTPUT AS 5
i% = FREEFILE( )
```

The lowest available file number in this example is 2 because 1 is already being used, therefore **i** is set to 2.

{button ,AL(`OPEN_APPEND;OPEN_INPUT;OPEN_OUTPUT;;;',0,"Defaultoverview",)} Related Topics

GETAPPHANDLE function

ReturnValue& = GETAPPHANDLE ( )

Returns the Application Instance Handle for the Corel application that is running a script. For example, if you are running the script from the Corel SCRIPT Editor, GETAPPHANDLE returns the Editor's Application Instance Handle. If you run a script from Corel VENTURA, GETAPPHANDLE returns VENTURA's Application Instance Handle. This function is used in conjunction with DLL calls that require the application's handle.

Return Value

Lets you specify a numeric variable that is passed the Application Instance Handle.

Example

```
hand = GETAPPHANDLE()
```

{button ,AL(`declare_lib;open_output;open_append;open_input;getwinhandle;getapphandle;',0,"Defaultoverview",)}
Related Topics

GETCOMMANDLINE function

ReturnValue$ = GETCOMMANDLINE ( )

Returns the parameters used in the command line that launch a script. To test this command, specify a command line in the Command Line text box in the Corel SCRIPT Editor's Options dialog box (click Tools, Options, Environment tab). For more information about command lines, click  .

Return Value

String variable that is passed the command line parameters.

Example

```
CommandLine$ = GETCOMMANDLINE ( )
```

{button ,AL(`setcurrfolder;getCURRFOLDER;getfolder;ht_start_cse_custom;;',0,"Defaultoverview",)} Related Topics

GETCURRFOLDER function

ReturnValue$ = GETCURRFOLDER ( )

Returns the name of the active Windows folder and path.

Return Value

String variable that is passed the name of the active Windows folder and path.

Note

- You can set the active folder using the **SETCURRFOLDER** statement.
- In Corel SCRIPT version 7.0, the **GETCURRFOLDER** function and the **SETCURRFOLDER** statement replace the **CURRFOLDER** statement.

Example

```
SETCURRFOLDER "c:\corel\graphics8\scripts\"
folder$ = GETCURRFOLDER ( )
MyFile$ = "\MyScript.csc"
MyPathFile = folder$ & MyFile&
```

In the above example the first line sets the active folder. The second line assigns the active folder to a string variable. The third line assigns a file name to a string variable. In the last line a string variable is assigned a value which is made by combining the folder and file string variables.

{button ,AL(`setcurrfolder;getCURRFOLDER;getfolder;;;',0,"Defaultoverview",)} Related Topics

GETPROCESSINFO function

ReturnValue& **= GETPROCESSINFO (ProcessHandle)**

This function returns the status of an executable.

Return Value

Numeric variable that is passed a value that indicates whether an executable is running. If the executable is running, this variable is passed the value 259.

Parameter	Description
ProcessHandle	Numeric expression specifying the Windows Process Handle of an executable. Use the **STARTPROCESS** function to determine an executable's Windows Process Handle.

Example

```
launch = STARTPROCESS ("C:\WINDOWS\CALC.EXE")
... 'other script statements
... 'other script statements
... 'other script statements
Calc_Status = GETPROCESSINFO (launch)
```

The above example launches the Windows Calculator and passes the Windows Process Handle to the **launch** variable. The **launch** variable is used with the **GETPROCESSINFO** function to determine whether the Calculator is running.

{button ,AL(`STARTPROCESS ;GETPROCESSINFO;INCLUDE;INPUT;cs_exe_dll;;;',0,"Defaultoverview",)} Related Topics

GETSCRIPTFOLDER function

ReturnValue$ = GETSCRIPTFOLDER ( )

Returns the path and the folder where the executing script resides. If this function is used in a Corel SCRIPT Executable, it returns the path and folder where the Executable resides.

Return Value

String variable that is passed the path and folder of an executing script or Executable.

Note

- If the script has not been previously saved to disk or a network, this function returns an empty string.

Example

```
folder = GETSCRIPTFOLDER ( )
```

{button ,AL(`DECLARE_LIB;OPEN_OUTPUT;OPEN_INPUT;GETWINHANDLE;GETAPPHANDLE;Creating Corel SCRIPT Executables;;;',0,"Defaultoverview",)} Related Topics

GETTEMPFOLDER function

ReturnValue$ = GETTEMPFOLDER ( )

Returns the path and the folder of the system's Windows temporary folder.

Return Value

String variable that is passed the path and folder of the system's Windows temporary folder.

Example

```
t_folder = GETTEMPFOLDER ( )
```

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

GETVERSION function

ReturnValue& **= GETVERSION (option)**

Returns the system or Corel SCRIPT version numbers.

Return Value

The GETVERSION function returns one of the following values:

- **Option 0** - Corel SCRIPT run-time interpreter:

  Four-digit number. The first two digits represent the major version number and the last two digits represent the minor version number. For example, the four digit number 7001 indicates major version 7.0 and minor version 01.

- **Option 10** - Corel SCRIPT compiler version:

  Four-digit number. The first two digits represent the major version number and the last two digits represent the minor version number. For example, the four digit number 7001 indicates major version 7.0 and minor version 01.

- **Option 30** - Windows platform:

  Single-digit number indicating the Windows platform that Corel SCRIPT is being used with.

  0 Win32s (Windows 3.11)

  1 Windows 95

  2 Windows NT

- **Option 31** - Windows major version number:

  For example, if you're running Windows 95 and your Windows version number is 4.00.950, the major version number is 4. The same applies to Windows NT.

- **Option 32** - Windows minor version number:

  For example, if you're running Windows 95 and your Windows version number is 4.00.950, the minor version number is 0.

- **Option 33** - Windows build number:

  For example, if you're running Windows 95 and your Windows version number is 4.00.950, the minor version number is 950.

Parameter	Description
option	Lets you specify the system or Corel SCRIPT component to query: 0 Corel SCRIPT run-time interpreter version (the SCINT**xx**.DLL file being used with the current session of Corel SCRIPT) 10 Corel SCRIPT compiler version number 30 Windows platform 31 Windows major version number 32 Windows minor version number 33 Windows build number

Note

- For a script, Executable, DLL, or Corel Add-on created with Corel SCRIPT to run, the major version numbers for Corel SCRIPT (the compiler) and the Corel SCRIPT run-time interpreter (SCINT**xx**.DLL) must be the same or else an error will occur. A difference in the minor version numbers will not cause an error.

Example

```
CS_version = GETVERSION(10)
MESSAGE "Corel SCRIPT major version number " & LEFT (CS_version, 2)
MESSAGE "Corel SCRIPT minor version number " & RIGHT (CS_version, 2)
```

In the above example, the first line passes the Corel SCRIPT version number to **CS_version**. A message box is then used to display the first two digits in CS_version using the **LEFT** function. Next, a message box is used to display the first two digits in CS_version using the **RIGHT** function.

{button ,AL(`Creating Corel SCRIPT Executables;getscriptfolder;getapphandle;getwinhandle;;',0,"Defaultoverview",)}
Related Topics

GETWINHANDLE function

ReturnValue& = GETWINHANDLE ( )

Returns the window handle for the window that is running the script. For example, if you are running the script from the Corel SCRIPT Editor, **GETWINHANDLE** returns the Editor's Windows handle. If you run a script from CorelDRAW, **GETWINHANDLE** returns DRAW's Windows handle. This function is used in conjunction with DLL calls that require the window's handle.

Return Value

Lets you specify a numeric variable that is passed the window's handle.

Note

- If you run a Corel SCRIPT Executable, the **GETWINHANDLE** function returns 0. For more information about Corel SCRIPT Executables, click  .

Example

```
hand = GETWINHANDLE()
```

{button ,AL(`declare_lib;open_output;open_append;open_input;getwinhandle;getapphandle;',0,"Defaultoverview",)}
Related Topics

INCLUDE statement

#INCLUDE filename

Lets you specify a Corel SCRIPT script to execute from the executing script. The specified script is treated as having its contents inserted into the executing script at the line holding the **INCLUDE** statement.

If you re-use many of the same constant, variable, and procedure declarations, you should consider putting that information in a separate script. Keeping this information in a separate script allows you to type the information once, and then call it as many times as you need with an INCLUDE statement in any new script you create.

For example, if you use the **WITHOBJECT** statement to call CorelDRAW 7or Corel VENTURA 7, you can insert the following statements in a INCLUDE file:

```
GLOBAL CONST DRAW7 = "CorelDraw.Automation.7"
GLOBAL CONST VENTURA7 = "CorelVentura.Automation.7"
```

The two statements above create two global constants named **DRAW6** and **VENTURA7**. If you always convert from tenths of a micron to another unit of measurement, you could include the following statement:

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
```

The **LENGTHCONVERT** statement creates a variable (**M_POINT**) that is equal to the number of tenths of a micron in a point.

Parameter	Description
filename	String expression specifying the filename, and optionally the path. If the path is not specified in **filename**, Corel SCRIPT will search for the file on your system in the following manner: 1 The active folder. You can use the **GETCURRFOLDER** statement to set or determine the active folder. 2 Folders in the path. The path is specified in the systems AUTOEXEC.BAT file. 3 Folders set in the Corel SCRIPT Editor's INCLUDE option. Click >> for more information about setting INCLUDE folders. 4 Folders specified with the **ADDFOL** statement.

Note

- Corel SCRIPT 7 introduces and includes CSI files. CSI are scripts (text files) that define commonly used constants. These files can be edited in the Corel SCRIPT Editor, and each Corel application that supports Corel SCRIPT has its own CSI file.

  Based on a typical Corel installation, an application's CSI file resides in the **C:\COREL*CorelSuite**application*\SCRIPTS** folder, where ***CorelSuite*** refers to the Corel products installed and ***application*** refers to the Corel application's folder. For example, the CorelDRAW 7 CSI file may reside in **C:\COREL\DRAW70\DRAW\SCRIPTS** folder and Corel VENTURA 7 CSI file may reside in the **C:\COREL\VENTURA7\VENTURA\SCRIPTS** folder.

  A general constants CSI file (SCPCONST.CSI) for Corel SCRIPT normally resides in the **C:\COREL*CorelSuite*\SCRIPTS** folder.

- The pound sign (#) is required in the syntax.

Example

```
#INCLUDE "My_constants.CSC"
#INCLUDE "SCPCONST.CSI"
```

The above example includes the Corel SCRIPT script MY_CONSTANTS.CSC and the SCPCONST constants file in the executing script.

{button ,AL(`GETCURRFOLDER;lengthconvert;angleconvert;addfol;const;;',0,"Defaultoverview",)} Related Topics

KILL statement

KILL fileName

Deletes a file. This statement is the same as clicking File, Delete in the Windows Explorer or in My Computer in Windows 95.

Parameter	Description
fileName	String expression specifying the filename to delete. You can use wild cards (* and ?) if you want to delete a group of files. For example, **script*.*** deletes all the files in the current folder beginning with **script**. Using **script?.*** deletes all the files in the current folder that begin with **script** and are followed by only one more character.

Note

- An open file cannot be deleted.

Example

```
KILL "temp.out"
```

Deletes the file TEMP.OUT in the current folder.

```
KILL "C:\MyDocs\temp.out"
```

Deletes the file TEMP.OUT in the **C:\MyDocs** folder.

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

MKFOLDER statement and function

Statement: MKFOLDER folderName

Function: ReturnValue& = MKFOLDER (folderName)

Creates a new folder.

Return Value

The MKFOLDER function returns one of the following values:

- TRUE (-1) the folder was created
- FALSE (0) the folder was not created

Parameter	Description
folderName	String expression specifying the name of the folder to be created. Path information is optional.

Example

```
MKFOLDER "work"
```

Creates the folder **work** as a subfolder of the current folder.

```
success = MKFOLDER ("work")
```

Creates the folder **work** as a subfolder of the current folder and assigns -1 to **success**.

{button ,AL(`;GETCURRFOLDER;SETCURRFOLDER;RMFOLDER;;;;',0,"Defaultoverview",)} Related Topics

REGISTRYQUERY function

ReturnValue = REGISTRYQUERY (MainKey, SubKey, Value)

Returns the value data of a specified value key in the system's Windows registry. This function can help you determine where programs and files are installed on a user's system. This type of information is important when creating scripts that are to run on different system setups.

Return Value

Lets you specify the variable that is passed the value data of a specified value key in the Windows registry. Since this function can pass a string or numeric value, the variable you specify should be a variant. You can use the **GETTYPE** function to determine a variant's subtype.

Parameter	Description
MainKey	Lets you specify the main registry value key to query: 0 HKEY_CLASSES_ROOT 1 HKEY_CURRENT_USER 2 HKEY_LOCAL_MACHINE 3 HKEY_USERS 4 HKEY_PERFORMANCE_DATA 5 HKEY_CURRENT_CONFIG 6 HKEY_DYN_DATA
SubKey	Lets you specify the sub registry value key to query. This must be a complete key path. In Windows 95 for example, "SOFTWARE\Microsoft\Windows\CurrentVersion\Fonts" is a complete path.
Value	Lets you specify the registry value key to query. To specify a default value, use an empty string. Specify an empty string by using two quotation marks ("").

Note

- You cannot use this command to query binary values except those that can be converted to a number.

Example

```
Config_Ventura = REGISTRYQUERY (2, "SOFTWARE\Corel\Corel Ventura\7.0", "ConfigDir")
```

The above example returns the root folder where Corel VENTURA 7 is installed.

```
Arial_file = REGISTRYQUERY (2, "SOFTWARE\Microsoft\Windows\CurrentVersion\Fonts", "Arial
(TrueType)")
```

The above example returns the Arial True Type font's file name.

```
YourName$ = REGISTRYQUERY(2,"SOFTWARE\Corel","UserName")
```

The above example returns the name of the registered owner of Corel Software.

```
CompanyName$ = REGISTRYQUERY(2,"SOFTWARE\Corel","ORGANIZATION")
```

The above example returns the organization name of the registered owner of Corel Software.

```
Phone$ = REGISTRYQUERY(2,"SOFTWARE\Corel","PHONENUMBER")
```

The above example returns the phone number of the registered owner of Corel Software

{button ,AL(`GETSCRIPTFOLDER;GETAPPHANDLE;GETWINHANDLE;GETTYPE;;',0,"Defaultoverview",)} Related Topics

RENAME statement and function

RENAME file_folder1, file_folder2, overwrite

The RENAME statement changes the name of a file or folder, or can be used to move a file. You cannot move a folder using the RENAME statement.

You can also use RENAME as a function: it returns TRUE (-1) if the RENAME operation is successful, FALSE (0) if is not.

Parameter	Description
file_folder1	String expression specifying the name of the file or folder to move. **file_folder1** can include drive and folder path specifics ▸ if path specifics are not included, RENAME assumes the current folder.
file_folder2	String expression specifying the name of the file where **file_folder1** is to be moved. **file_folder2** can include drive and folder path specifics ▸ if path specifics are not included, RENAME assumes the current folder.
overwrite	If **file_folder2** already exists, determines whether to overwrite the existing file (you cannot overwrite existing folders): 0 = rename and overwrite 1 = overwrite fails (default if omitted)

Example

```
' statement example
DIM x AS STRING
DIM y AS STRING
x = "C:\work\example1.vp"
y = "D:\work\example1.vp"
RENAME x, y, 0
```

The above example moves the EXAMPLE1.VP file to the Work folder on the D drive.

```
' statement example
DIM x AS STRING
DIM y AS STRING
x = "C:\work\example1.cdr"
y = "C:\work\example2.cdr"
success = RENAME (x, y, 0)
```

The above example renames the EXAMPLE1.CDR file to EXAMPLE2.CDR, and assigns -1 to **success**.

{button ,AL(`FINDFIRSTFOLDER_FINDNEXTFOLDER;rename;copy;GETCURRFOLDER;rmfolder;SETcurrfolder;;',0,"Defaultoverview",)} Related Topics

RMFOLDER statement and function

RMFOLDER folderName

Removes an existing folder. The folder must be empty before it can be deleted. You can also use **RMFOLDER** as a function: it returns TRUE (-1) if the folder was removed, FALSE (0) if it was not.

Parameter	Description
folderName	String expression specifying the name of the folder to remove. **folderName** can include drive specifics ▶ if drive specifics are not included, RENAME assumes the current drive.

Example

```
RMFOLDER "C:\TEMP\WORK"
```

Removes the Work folder from the Temp folder.

```
success% = RMFOLDER "C:\TEMP\WORK"
```

Removes the Work folder from the Temp folder and assigns -1 to **success**.

{button ,AL(`KILL;MKFOLDER;GETCURRFOLDER;;;',0,"Defaultoverview",)} Related Topics

SETCURRFOLDER statement

SETCURRFOLDER FolderName

Sets the active Windows folder and path.

Parameter	Description
FolderName	String expression specifying a system folder and path.

Note

- In Corel SCRIPT version 7.0, the **SETCURRFOLDER** statement and the **GETCURRFOLDER** function replace the **CURRFOLDER** statement.

Example

```
SETCURRFOLDER "C:\corel\MyDocs\"
```

The above example sets the active folder and path to C:\COREL\MYDOCS\.

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

STARTPROCESS statement and function

Statement: STARTPROCESS exe

Function: ReturnValue& = STARTPROCESS (exe)

This statement or function launches executable files (.EXE files). You can also use this statement to launch Corel applications and Corel SCRIPT Executables.

Return Value

Numeric variable that is assigned a value indicating whether **STARTPROCESS** was not able to launch the specified executable. If the specified executable was not launched, 0 is assigned; otherwise the Windows Process Handle is returned.

Parameter	Description
exe	String expression specifying the executable to launch. The string expression should also include the executable's path and file extension.

Note

- Use the **GETPROCESSINFO** function to determine whether an executable is still running.

Example

```
STARTPROCESS "C:\WINDOWS\CALC.EXE"
```

The above example attempts to launch the Windows Calculator.

```
launch = STARTPROCESS ("C:\WINDOWS\CALC.EXE")
```

The above example attempts to launch the Windows Calculator and assigns a value to the **launch** variable, indicating whether the calculator was launched.

{button ,AL(`STARTPROCESS ;GETPROCESSINFO;INCLUDE;INPUT;cs_exe_dll;;;',0,"Defaultoverview",)} Related Topics

File Number

An integer (whole number) value between 1-10, inclusive. Non-integers are truncated.

CLOSE statement

CLOSE #num,...

Closes a text file opened with an OPEN statement (**OPEN...APPEND** or **OPEN...OUTPUT**).

Parameter	Description
num	Numeric expression(s) specifying a file number to close. If not specified, all open files are closed. The # sign is optional.

Note

- In your scripts, every OPEN statement should have a corresponding CLOSE statement.

Examples

```
CLOSE
```
Closes all open files.

```
CLOSE #1
```
Closes the file opened as 1.

```
CLOSE 1
```
Closes the file opened as 1.

```
CLOSE 1, 3, 5
```
Closes the files opened as 1, 3, and 5.

{button ,AL(`open_append;open_input;open_output;;;',0,"Defaultoverview",)} Related Topics

EOF function

EOF (#num)

Returns TRUE (-1) if the file pointer is at the end of an open text file in your computer's memory. Returns FALSE (0) if the file contains data beyond the pointer. The statement is often used to determine whether to continue processing a file.

Parameter	Description
#**num**	Numeric expression specifying an open file number to examine. The # sign is optional.

Example

```
IF EOF(1) THEN CLOSE 1
```

If the pointer is at the end of the file, then close the file.

{button ,AL(`lof;seek;open_append;print;write;',0,"Defaultoverview",)} Related Topics

INPUT function

INPUT(bytes, #num)

Starting at the pointer, reads a number of bytes (characters) from a text file.

Parameter	Description
bytes	Numeric expression specifying the number of bytes to be read.
#**num**	Numeric expression specifying an open file number to examine. The # sign is optional.

Example

```
myExtract$ = INPUT(50, #1)
```

Reads 50 characters from file 1 and assigns them to the variable **myExtract$**.

{button ,AL(`INPUT;LINE_INPUT;SEEK;FILEPOS;EOF;LOF;',0,"Defaultoverview",`main')} Related Topics

INPUT # statement

INPUT #num, var1, var2, ...

Reads from a file to a list of variables. Values in the file are separated by commas. Character strings that include commas must be enclosed in quotation marks. The quotation marks do not appear in the variable.

Parameter	Description
#**num**	Numeric expression specifying an open file number to examine.
var1, var2, ...	The variables that receive the fields as they are read from the file.

Notes

- The number sign (#) is required in the syntax.
- Numeric data with decimals can be input only if the period (**.**) is used as the decimal separator.
- If you're converting dates, they must be in the standard International format (yy/MM/dd hh:mm:ss).
- Booleans can be input only as TRUE or FALSE.

Example

```
INPUT #1, title$, number%
```

Reads a string and a numeric variable from file 1. Assigns the values to the string variable **title$** and the integer variable **number%**. The contents of file 1 could be in either of two formats:

```
"Ottawa", 50
"Huckleberry Finn", 101
"Hamlet", 220
```

or

```
"Ottawa", 50, "Huckleberry Finn", 101, "Hamlet", 220
```

{button ,AL(`input_dollar;open_input;line_input;write;;',0,"Defaultoverview",)} **Related Topics**

LINE INPUT statement

LINE INPUT #num, string

Reads the next line from an open text file in your computer's memory into a string.

Parameter	Description
#num	Numeric expression specifying an open file number to examine.
string$	String expression specifying the string variable to hold the line from the file.

Note

- The number sign (#) is required in the syntax.

Example

```
LINE INPUT #1, MyString$
```

Reads the next line from file 1 and places the string in the variable **MyString$**.

{button ,AL(`input_dollar;input;open_input;line_input;;;',0,"Defaultoverview",)} Related Topics

LOF function

LOF(#num)

LOF (Length of File) returns the number of bytes in an open text file.

Parameter	Description
#**num**	Numeric expression specifying an open file number to examine. The # sign is optional.

Example

```
bytes% = LOF(1)
```

Sets **bytes** to the number of bytes in file 1.

{button ,AL(`eof;seek;open_append;print;write;',0,"Defaultoverview",)} **Related Topics**

OPEN...APPEND statement

OPEN fileName FOR APPEND AS #num

Opens a text file to add or append sequential data to the end of the file. Data is appended using the **PRINT** or **WRITE** statement. The FOR clause is required.

Parameter	Description
fileName	String expression specifying a filename to open. If the file doesn't exist, it is created and then opened. Drive and folder information is optional.
#**num**	Numeric expression specifying a file number to associate with the open file. The # sign is optional.

Note

- You cannot use the **COPY** or **RENAME** command on an open file.

Example

```
OPEN "oldfile" FOR APPEND AS 3
```

Opens the file OLDFILE as file number 3 to append sequential data.

{button ,AL(`input_dollar;open_append;open_input;open_output;ACCESS;CLOSE;EOF;FILEMODE;FILEPOS;freefile;input;line_input;lof;print;seek;write',0,"Defaultoverview",`main')} Related Topics

OPEN...INPUT statement

OPEN fileName FOR INPUT AS #num

Opens a text file so that data can be read from it using the **INPUT** or **LINE INPUT** statement.

Parameter	Description
fileName	String expression specifying a filename to open. An error occurs if the file doesn't exist. Drive and folder information is optional.
#**num**	Numeric expression specifying a file number to associate with the open file. The # sign is optional.

Note

- You cannot use the **COPY** or **RENAME** command on an open file.

Example

```
OPEN FileString$ FOR INPUT AS 2
```

Opens the file specified by **FileString$** for input.

{button ,AL(`input_dollar;open_append;open_input;open_output;ACCESS;CLOSE;EOF;FILEMODE;FILEPOS;freefile;input;line_input;lof;print;seek;write',0,"Defaultoverview",`main')} Related Topics

OPEN...OUTPUT statement

OPEN fileName$ FOR OUTPUT AS #num%

Opens a text file so that data can be written into it using the **PRINT** or **WRITE** statement. The FOR clause is required.

Parameter	Description
fileName	String expression specifying a filename to open. If the file doesn't exist, it is created and then opened. Drive and folder information is optional.
#**num**	Numeric expression specifying a file number to associate with the open file. The # sign is optional.

Note

- You cannot use the **COPY** or **RENAME** command on an open file.

Example

```
OPEN "c:\temp\workfile.tmp" FOR OUTPUT AS 1
```

Opens the file C:\Temp\WORKFILE.TMP for output.

{button ,AL(`input_dollar;open_append;open_input;open_output;ACCESS;CLOSE;EOF;FILEMODE;FILEPOS;freefile;input;line_input;lof;print;seek;write',0,"Defaultoverview",`main')} Related Topics

PRINT statement

PRINT #num, expression1 { , | ; } expression2 { , | ; } ...

Prints expression(s) to an open text file (**OPEN...APPEND** or **OPEN...OUTPUT**) in your computer's memory.

Parameter	Description
#num	Numeric expression specifying a file number to print to.
expression1, expression2,...	The numeric or string expressions to print to the specified file.
{ , | ; }	Lets you specify how to separate the expressions being printed to the specified file. The comma places a tab between expressions. The semi-colon does not use a space, or a comma, between printed expressions.

Notes

- Ending a **PRINT** statement without a comma or a semicolon inserts a line return at the end of the printed expression.
- Strings are not enclosed in quotation marks when printed.
- Numeric expressions have either a leading space or a negative sign.
- The number sign (#) is required in the syntax.
- Dates and time are printed using the system's regional settings. The decimal character is also dependent on the system's regional settings. In Windows 95, click Start, Setting, Control Panel, Regional settings to view and change your system settings.
- Booleans are printed as TRUE or FALSE.

Example

```
PRINT #1, "A",          'inserts a tab after A
PRINT #1, "B"           'inserts a return after B
PRINT #1, "C";          'no spacing after C
PRINT #1, "D"
```

The above example prints the following to file #1 (there is a tab between A and B):

```
A   B
CD
```

In the following example, a blank line is printed, and then the string **HEADING:** is followed by the value of **string1$**, a comma, a space, and the value in **my_int%**.

```
PRINT #1,
PRINT #1, "HEADING: "; string1$; ", "; my_int%
```

{button ,AL(`seek;print;write;open_append;open_output;input;;',0,"Defaultoverview",)} Related Topics

SEEK statement

SEEK #num, position

To prepare for subsequent input, moves the file pointer to a specific byte position in an open text file in your computer's memory. Characters, spaces, tabs, and line breaks, are counted as characters.

Parameter	Description
num	Numeric expression specifying a file number to examine. The # sign is optional.
position	Numeric expression specifying the byte position to SEEK (1 is 1st byte).

Note

- The seek always starts from the first position in the file.
- The seek range is limited from 1 to length of the file (**LOF**) + 1.

Example

```
SEEK 1, 100
```

Moves the file pointer to the 100th byte in file 1.

{button ,AL(`OPEN_APPEND;OPEN_INPUT;OPEN_OUTPUT;lof;;',0,"Defaultoverview",)} Related Topics

WRITE statement

WRITE #num, expression1 { , | ; } expression2 { , | ; } ...

Writes expressions to an open text file (**OPEN...APPEND** or **OPEN...OUTPUT**) in your computer's memory.

Parameter	Description
#num%	Numeric expression specifying a file number to write to.
expression1, expression2,...	The numeric or string expressions to print to the specified file.
{ , | ; }	Lets you specify a separator between expressions. Both the comma and the semicolon insert a comma as a separator.

Notes

- Ending a complete **WRITE** statement without a comma or a semicolon inserts a line return at the end of the written expression.
- Strings are enclosed in quotation marks when printed.
- Numeric expressions have either a leading space or a negative sign.
- The number sign (#) is required in the syntax.
- Numeric data is written using the period (**.**) as a decimal separator.
- If you're converting dates, they must be in the standard International format (yy/MM/dd hh:mm:ss).
- Booleans are written as TRUE or FALSE.

Example

```
WRITE #1, "A ",          'inserts a space and comma after A
WRITE #1, "B"            'inserts a return after B
WRITE #1, "C";           'inserts a comma after C
WRITE #1, "D"            'inserts a return after D
WRITE #1,                'prints a blank line
```

The above example prints the following to file #1:

```
"A ","B"
"C","D"
```

{button ,AL(`seek;print;write;open_append;open_output;input;;',0,"Defaultoverview",)} Related Topics

ABS function

ABS(x)

Returns the absolute value of a number. Absolute value is the positive value of a number.

Parameter	Description
x	Any numeric expression.

Examples

```
x% = ABS(-3)
y = ABS(3)
```

Both the above examples return 3 to **x** and **y**.

{button ,AL(`abs;sgn;fix;int;;',0,"Defaultoverview",)} Related Topics

ACOS function

ACOS(x)

Returns the inverse cosine (arc cosine) of a given value. The result is an angle measured in radians between 0 and π (where π is approximately 3.14152).

Parameter	Description
x	A numeric expressionbetween -1 and 1.

Note

- To convert the result from radians to degrees, use the **ANGLECONVERT** function or multiply the result by 180/3.14152.

Examples

```
v = ACOS(-0.75)
w = ACOS(0.75)
x = ACOS(0)
y = ACOS(1)
```

In the above example, **v** is equal to 2.418858406, **w** is equal to 0.722734248, **x** is equal to 1.570796327, and **y** is equal to 0.

{button ,AL(`Math_PASTE;ACOS;ASIN;ATAN;;',0,"Defaultoverview",)} Related Topics

ASIN function

ASIN(x)

Returns the inverse sine (arc sine) of a given value. The result is an angle in radians bounded by -p /2 and p /2.

Parameter	Description
x	A numeric expressionbetween -1 and 1.

Note

- To convert the result from radians to degrees, use the **ANGLECONVERT** function or multiply the result by 180/3.14152.

Examples

```
w = ASIN(-0.75)
x = ASIN(0.75)
y = ASIN(0)
z = ASIN(1)
```

In the above example, **w** is equal to -0.8480621, **x** is equal to 0.8480621, **y** is equal to 0, and **z** is equal to 1.570796.

{button ,AL(`Math_PASTE;ACOS;ASIN;ATAN;;',0,"Defaultoverview",)} Related Topics

ATAN function

ATAN(x)

Returns the inverse tangent (arc tangent) of a given value. The result is an angle bounded by -π /2 (-90 degrees) and π /2 (90 degrees) measured in radians.

Parameter	Description
x	A numeric expression.

Note

- To convert the result from radians to degrees, use the **ANGLECONVERT** function or multiply the result by 180/3.14152.

Examples

```
w = ATAN(-0.75)
x = ATAN(0.75)
y = ATAN(0)
z = ATAN(1)
```

In the above example, **w** is equal to -0.6435011, **x** is equal to 0.6435011, **y** is equal to 0, and **z** is equal to 0.7853982.

{button ,AL(`Math_PASTE;ACOS;ASIN;ATAN;;',0,"Defaultoverview",)} Related Topics

COS function

COS(x)

Returns the cosine of an angle measured in radians.

Parameter	Description
x	Any numeric expression. Lets you specify the angle measured in radians.

Notes

- The result of COS is between -1 and 1.
- To convert degrees to radians, multiply degrees by 3.14159/180 (π is approximately equal to 3.14159) or use the **ANGLECONVERT** function.

Examples

```
degreeMeasure% = 45
MyResult = COS(3.14159/180*degreeMeasure%)
```

The above example returns the COS of 45 degrees as expressed in radians. The variable **MyResult** equals 0.707106781.

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

DEC function

DEC(x)

Returns the conversion of a hexadecimal value into decimal notation (as a long data type).

Parameter	Description
x	A string expression representing a hexadecimal number.

Notes

- Decimal notation is a numerical system based on groups of ten units.
- The highest value you can convert is 7FFFFFFF.
- The **HEX** function performs the opposite conversion, from decimal to hexadecimal.

Examples

```
x& = DEC("A27")
```

In the above example, **x** equals 2599.

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

EXP function

EXP(x)

Raises *e* to a given exponent where *e* is the base of the natural logarithm which equals 2.718281828.

Parameter	Description
x	Any numeric expression.

Note

- EXP is the inverse of the natural logarithm (**LN**).

Example

```
x = EXP(7.89)
```

In the above example, **x** equals 2670.444.

{button ,AL(`log;ln;exp;;;',0,"Defaultoverview",)} Related Topics

FIX function

FIX(x)

Removes an argument's decimal or fraction and rounds towards 0. An integer is returned.

Parameter	Description
x	A numeric expression.

Note

- Both **INT** and FIX return the integer portion of a given number. However, INT returns the greatest integer less than or equal to the number, while FIX returns the integer portion given, without any decimal points represented. As a result, -5.26 becomes -5 under the FIX function, and -6 under the INT function.

Examples

```
vv = FIX(12.65)
yy = FIX(-47.29)
```

In the above example, **vv** equals 12 and **yy** equals -47.

```
v = INT(12.65)
y = INT(-47.29)
```

In the above example, **v** equals 12 and **y** equals -48.

{button ,AL(`abs;sgn;fix;int;;',0,"Defaultoverview",)} Related Topics

HEX function

HEX(x)

Converts a number to its corresponding hexadecimal string value.

Parameter	Description
x	A numeric expressionspecifying a decimal value to convert.

Notes

- Decimal numbers are rounded to the nearest whole number before being converted.
- Hexadecimal notation is a numerical system based on groups of sixteen units. Hexadecimal numbers can be expressed in Corel SCRIPT by using the prefix **&h**. For example, &h10 or &h1ABC.
- The **DEC** function performs the opposite conversion, from hexadecimal to decimal.

Example

```
x$ = HEX(27)
x$ = HEX(27.25)
```

In the above example **x** and **y** are both passed the value "1B".

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

INT function

INT(x)

Removes an argument's decimal or fraction and rounds down to the nearest integer. An integer is returned.

Parameter	Description
x	A numeric expression.

Note

- Both INT and **FIX** return the integer portion of a given number. However, INT returns the greatest integer less than or equal to the number, while FIX returns the integer portion given, without any decimal points represented. As a result, -5.26 becomes -5 under the FIX function, and -6 under the INT function.

Examples

```
v = INT(12.65)
y = INT(-47.29)
```

In the above example, **v** equals 12 and **y** equals -48.

```
vv = FIX(12.65)
yy = FIX(-47.29)
```

In the above example, **vv** equals 12 and **yy** equals -47.

{button ,AL(`abs;sgn;fix;int;;',0,"Defaultoverview",)} Related Topics

LN function

LN(x)

Returns the natural logarithm (base *e*) of a number.

Parameter	Description
x	Any positive numeric expression.

Notes

- LN is the inverse of the **EXP** function; therefore LN(EXP(**x**)) equals **x**.
- Natural logarithms are based on the constant *e* which is equal to 2.718281828.

Examples

```
w = LN(1.2)
x = LN(30)
y = LN(1)
z = LN(EXP(30))
```

In the above example, **w** is equal to 0.1823215568, **x** is equal to 3.401197382, **y** is equal to 0, and **z** is equal to 30.

{button ,AL(`log;ln;exp;;;',0,"Defaultoverview",)} Related Topics

LOG function

LOG(x)

Returns the base-10 logarithm of a number.

Parameter	Description
x	Any positive numeric expression.

Note

- The LOG function uses a base of 10. If another base is needed, use LOG(x)/LOG(b) formula where **b** is the base.

Examples

```
x = LOG(25)
y = LOG(5)
```

In the above example, **x** equals 1.397940 and **y** equals 0.6989700.

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

RANDOMIZE function

RANDOMIZE x

Sets the random number generator seed to the integer portion of the argument value.

Parameter	Description
x	Any numeric expression. This is an optional parameter. **Randomize** uses the argument as a seed to start a new sequence of random numbers. Using RANDOMIZE without an argument initializes the seed to the system timer. If RANDOMIZE isn't used before calling **RND**, the same sequence of numbers will result every time the random number generator is used.

Examples

```
RANDOMIZE
RANDOMIZE 24
```

In the above example, the first line initializes the random number generator seed to the system timer. The second line uses 24 as the first seed in the random number generator's sequence.

The following example returns 5 random integers between 1 and 10 to the variable **a_random**. Each random value is also displayed in a message box.

```
RANDOMIZE
FOR x = 1 TO 5
   lower=1
   upper=10
   a_random = INT((upper - lower +1)*RND()+lower)
   MESSAGE a_random
NEXT x
```

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

RND function

RND(x**)**

Returns a random number.

Parameter	Description
x	A numeric expression that determines the bounds of the random number. See the table below.

x's value	Random Number Bounds
Greater than 0	A number between 0 (lower bound) and x (upper bound).
Less than 0	A number between x (lower bound) and 0 (upper bound).
Omitted	A number between 0 and 1.

Note

- If **RANDOMIZE** isn't used before calling **RND** for the first time, the same sequence of numbers will result every time the script is run.

Examples

```
x = RND()
y = RND(-7)
z = RND(24)
```

In the above example, **x** returns a random number between 0 and 1, **y** returns a random number between -7 and 0, and **z** returns a random number between 0 and 24.

To create a random integer in a specified range, use the following formula:

```
INT((upper - lower + 1)*RND()+ lower)
```

where **upper** is the highest number in the specified range and **lower** is the lowest number in the specified range. The following example returns a random integer between 1 and 10 to the variable **a_random**.

```
lower = 1
upper = 10
a_random = INT((upper - lower +1)*RND()+lower)
```

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

SGN function

SGN(x)

Determines the sign (+ or -) of a number. Returns -1 if the number is negative, 1 if the number is positive, and 0 if the number is 0.

Parameter	Description
x	Any numeric expression.

Examples

```
x% = SGN(5)
y = SGN(0)
z = SGN(-0.021)
```

In the above example, **x** is equal to 1, **y** is equal to 0, and **z** is equal to -1.

{button ,AL(`abs;sgn;fix;int;;',0,"Defaultoverview",)} Related Topics

SIN function

SIN(x)

Returns the sine of an angle measured in radians.

Parameter	Description
x	Any numeric expression. Lets you specify the angle measured in radians.

Note

- The result of SIN is between -1 to 1, inclusive.
- To convert degrees to radians, multiply degrees by 3.14159/180 (π is approximately equal to 3.14159) or use the **ANGLECONVERT** function.

Examples

```
degreeMeasure% = 45
MyResult = SIN(3.14159/180*degreeMeasure%)
```

The above example returns the SIN of 45 degrees. The variable **MyResult** equals 0.70710678.

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

SQR function

SQR(x)

Returns the positive square root of a number.

Parameter	Description
x	Any non-negative numeric expression.

Examples

```
w = SQR(4)
x = SQR(0.175)
```

In this example, **w** is equal to 2 and **x** is equal to 0.4183300133.

{button ,AL(`abs;sgn;fix;int;;',0,"Defaultoverview",)} Related Topics

TAN function

TAN(x)

Returns the tangent of an angle measured in radians.

Parameter	Description
x	Any numeric expression. Lets you specify the angle measured in radians.

Note

- To convert degrees to radians, multiply degrees by 3.14159/180 (*p* • is approximately equal to 3.14159) or use the **ANGLECONVERT** function.

Example

```
radianMeasure = 0.5
MyResult = TAN(radianMeasure)
```

The above example returns the TAN of 0.5 radians. The variable **MyResult** equals 0.546302.

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

A

B

C

CreateBackground (PHOTO-PAINT)

D

DuotoneHandle (PHOTO-PAINT)
DuotoneInfo (PHOTO-PAINT)

E

EditCheckpoint (PHOTO-PAINT)
EditClear (PHOTO-PAINT)
EditClearClipboard (PHOTO-PAINT)
EditCopy (PHOTO-PAINT)
EditCopyToFile (PHOTO-PAINT)
EditCopyVisible (PHOTO-PAINT)
EditCut (PHOTO-PAINT)
EditCutIntoSelection (PHOTO-PAINT)
EditCutMask (PHOTO-PAINT)
EditCutSelection (PHOTO-PAINT)
EditFill (PHOTO-PAINT)
EditPasteDocument (PHOTO-PAINT)
EditPasteFromFile (PHOTO-PAINT)
EditPasteIntoSelection (PHOTO-PAINT)
EditPasteObject (PHOTO-PAINT)
EditPasteSelection (PHOTO-PAINT)
EditRestoreCheckpoint (PHOTO-PAINT)
EditSingleObject (PHOTO-PAINT)
Effect3DRotate (PHOTO-PAINT)
Effect3DStereoNoise (PHOTO-PAINT)
EffectAdaptiveUnsharp (PHOTO-PAINT)
EffectAddNoise (PHOTO-PAINT)
EffectAdjustBlur (PHOTO-PAINT)
EffectAdjustNoise (PHOTO-PAINT)
EffectAdjustSharpness (PHOTO-PAINT)
EffectBandPass (PHOTO-PAINT)
EffectBitPlanes (PHOTO-PAINT)
EffectBrickWall (PHOTO-PAINT)
EffectBubbles (PHOTO-PAINT)
EffectBumpMap (PHOTO-PAINT)
EffectCanvas (PHOTO-PAINT)
EffectCobblestone (PHOTO-PAINT)
EffectCrystalize (PHOTO-PAINT)
EffectDiffuse (PHOTO-PAINT)
EffectDirectionalSharpen (PHOTO-PAINT)
EffectDirectionalSmooth (PHOTO-PAINT)
EffectDisplace (PHOTO-PAINT)
EffectDustScratch (PHOTO-PAINT)
EffectEdgeDetect (PHOTO-PAINT)
EffectEmboss (PHOTO-PAINT)
EffectEtching (PHOTO-PAINT)
EffectFindEdges (PHOTO-PAINT)
EffectGaussianBlur (PHOTO-PAINT)
EffectGlass (PHOTO-PAINT)

EffectGlassBlock (PHOTO-PAINT)
EffectHalftone (PHOTO-PAINT)
EffectHighPass (PHOTO-PAINT)
EffectImpressionist (PHOTO-PAINT)
EffectJaggyDespeckle (PHOTO-PAINT)
EffectLensFlare (PHOTO-PAINT)
EffectLighting (PHOTO-PAINT)
EffectLocalHistogram (PHOTO-PAINT)
EffectLowPass (PHOTO-PAINT)
EffectMapToObject (PHOTO-PAINT)
EffectMaximum (PHOTO-PAINT)
EffectMedian (PHOTO-PAINT)
EffectMeshPoint (PHOTO-PAINT)
EffectMeshWarp (PHOTO-PAINT)
EffectMinimum (PHOTO-PAINT)
EffectMotionBlur (PHOTO-PAINT)
EffectOffset (PHOTO-PAINT)
EffectPageCurl (PHOTO-PAINT)
EffectPerspective (PHOTO-PAINT)
EffectPinchPunch (PHOTO-PAINT)
EffectPixelate (PHOTO-PAINT)
EffectPlastic (PHOTO-PAINT)
EffectPlugin (PHOTO-PAINT)
EffectPsychedelic (PHOTO-PAINT)
EffectPuzzle (PHOTO-PAINT)
EffectRadialBlur (PHOTO-PAINT)
EffectReliefSculpture (PHOTO-PAINT)
EffectRemoveMoire (PHOTO-PAINT)
EffectRemoveNoise (PHOTO-PAINT)
EffectRipple (PHOTO-PAINT)
EffectSharpen (PHOTO-PAINT)
EffectShear (PHOTO-PAINT)
EffectShearTable (PHOTO-PAINT)
EffectSmokedGlass (PHOTO-PAINT)
EffectSmooth (PHOTO-PAINT)
EffectSoften (PHOTO-PAINT)
EffectSolarize (PHOTO-PAINT)
EffectSwirl (PHOTO-PAINT)
EffectTheBoss (PHOTO-PAINT)
EffectTile (PHOTO-PAINT)
EffectTool (PHOTO-PAINT)
EffectTraceContour (PHOTO-PAINT)
EffectUnsharpMask (PHOTO-PAINT)
EffectUserDefined (PHOTO-PAINT)
EffectUserDefinedPoint (PHOTO-PAINT)
EffectVignette (PHOTO-PAINT)
EffectWetPaint (PHOTO-PAINT)
EffectWhirlpool (PHOTO-PAINT)
EffectWind (PHOTO-PAINT)
EffectZigZag (PHOTO-PAINT)

Ellipse (PHOTO-PAINT)
EllipseTool (PHOTO-PAINT)
EndAdjustEffect (PHOTO-PAINT)
EndColorEffect (PHOTO-PAINT)
EndColorMask (PHOTO-PAINT)
EndColorReplace (PHOTO-PAINT)
EndColorTable (PHOTO-PAINT)
EndConvertDuotone (PHOTO-PAINT)
EndConvertPaletted (PHOTO-PAINT)
EndDraw (PHOTO-PAINT)
EndEditFill (PHOTO-PAINT)
EndEffectLighting (PHOTO-PAINT)
EndEffectMeshWarp (PHOTO-PAINT)
EndEffectShear (PHOTO-PAINT)
EndEffectUserDefined (PHOTO-PAINT)
EndGuideline (PHOTO-PAINT)
EndImageEqualize (PHOTO-PAINT)
EndImageSTBalance (PHOTO-PAINT)
EndImageToneCurve (PHOTO-PAINT)
EndMovieFrameRate (PHOTO-PAINT)
EndObject (PHOTO-PAINT)
EndObjectColorTransparencyTool (PHOTO-PAINT)
EndObjectTagWWWURL (PHOTO-PAINT)
Eraser (PHOTO-PAINT)

F

FadeLastCommand (PHOTO-PAINT)
FileAcquireWithFile (PHOTO-PAINT)
FileClose (PHOTO-PAINT)
FileNew (PHOTO-PAINT)
FileOpen (PHOTO-PAINT)
FilePrint (PHOTO-PAINT)
FileRevert (PHOTO-PAINT)
FileSave (PHOTO-PAINT)
FileSelectPartialArea (PHOTO-PAINT)
Fill (PHOTO-PAINT)
FillBitmap (PHOTO-PAINT)
FillFountain (PHOTO-PAINT)
FillFountainApply (PHOTO-PAINT)
FillFountainColor (PHOTO-PAINT)
FillSolid (PHOTO-PAINT)
FillTexture (PHOTO-PAINT)
FillTextureSettings (PHOTO-PAINT)
FillTool (PHOTO-PAINT)
FilterGIF (PHOTO-PAINT)
FilterJPG (PHOTO-PAINT)
FilterOS2 (PHOTO-PAINT)
FilterPNG (PHOTO-PAINT)
FilterTGA (PHOTO-PAINT)
FilterWVL (PHOTO-PAINT)

G

GetChannelCount (PHOTO-PAINT)
GetChannelName (PHOTO-PAINT)
GetCurrentMovieFrame (PHOTO-PAINT)
GetDocumentCount (PHOTO-PAINT)
GetDocumentHeight (PHOTO-PAINT)
GetDocumentIsMovie (PHOTO-PAINT)
GetDocumentIsPartial (PHOTO-PAINT)
GetDocumentModified (PHOTO-PAINT)
GetDocumentName (PHOTO-PAINT)
GetDocumentType (PHOTO-PAINT)
GetDocumentWidth (PHOTO-PAINT)
GetDocumentXdpi (PHOTO-PAINT)
GetDocumentYdpi (PHOTO-PAINT)
GetFillColor (PHOTO-PAINT)
GetFrameCount (PHOTO-PAINT)
GetMaskPresent (PHOTO-PAINT)
GetMaskRectangle (PHOTO-PAINT)
GetObjectCount (PHOTO-PAINT)
GetObjectIsEditable (PHOTO-PAINT)
GetObjectIsSelected (PHOTO-PAINT)
GetObjectIsVisible (PHOTO-PAINT)
GetObjectMergeMode (PHOTO-PAINT)
GetObjectName (PHOTO-PAINT)
GetObjectOpacity (PHOTO-PAINT)
GetObjectProperties (PHOTO-PAINT)
GetObjectRectangle (PHOTO-PAINT)
GetPaintColor (PHOTO-PAINT)
GetPaintVersion (PHOTO-PAINT)
GetPaperColor (PHOTO-PAINT)
GetPartialDocumentHeight (PHOTO-PAINT)
GetPartialDocumentWidth (PHOTO-PAINT)
GetPhotoPaintDir (PHOTO-PAINT)
GetSelectedObjectsRectangle (PHOTO-PAINT)
Gradient (PHOTO-PAINT)
GradientPoint (PHOTO-PAINT)
GradientTool (PHOTO-PAINT)
GuidelineAdd (PHOTO-PAINT)
GuidelineDelete (PHOTO-PAINT)
GuidelineMove (PHOTO-PAINT)
GuidelineSelect (PHOTO-PAINT)

I

ImageApplyICCProfile (PHOTO-PAINT)
ImageAutoEqualize (PHOTO-PAINT)
ImageBCI (PHOTO-PAINT)
ImageColorBalance (PHOTO-PAINT)
ImageColorCrop (PHOTO-PAINT)
ImageColorTable (PHOTO-PAINT)
ImageColorTone (PHOTO-PAINT)

ImageConvert (PHOTO-PAINT)
ImageConvertDuotone (PHOTO-PAINT)
ImageConvertPaletted (PHOTO-PAINT)
ImageConvertVideoNTSC (PHOTO-PAINT)
ImageCrop (PHOTO-PAINT)
ImageCropToMask (PHOTO-PAINT)
ImageDeInterlace (PHOTO-PAINT)
ImageDesaturate (PHOTO-PAINT)
ImageDeskew (PHOTO-PAINT)
ImageDeskewCrop (PHOTO-PAINT)
ImageDuplicate (PHOTO-PAINT)
ImageEqualize (PHOTO-PAINT)
ImageEqualizeChannel (PHOTO-PAINT)
ImageFlipHorizontal (PHOTO-PAINT)
ImageFlipVertical (PHOTO-PAINT)
ImageGamma (PHOTO-PAINT)
ImageHSL (PHOTO-PAINT)
ImageHSLChannel (PHOTO-PAINT)
ImageInvert (PHOTO-PAINT)
ImageLevelThreshold (PHOTO-PAINT)
ImagePapersize (PHOTO-PAINT)
ImagePosterize (PHOTO-PAINT)
ImageReplaceColors (PHOTO-PAINT)
ImageResample (PHOTO-PAINT)
ImageRotate (PHOTO-PAINT)
ImageSelectiveColor (PHOTO-PAINT)
ImageSelectiveColorChannel (PHOTO-PAINT)
ImageSetChannel (PHOTO-PAINT)
ImageSplit (PHOTO-PAINT)
ImageSprayerSettings (PHOTO-PAINT)
ImageSprayerTool (PHOTO-PAINT)
ImageSTBalance (PHOTO-PAINT)
ImageSTColor (PHOTO-PAINT)
ImageToneChannel (PHOTO-PAINT)
ImageToneCurve (PHOTO-PAINT)
ImageToneCurve (PHOTO-PAINT)
ImageTonePoint (PHOTO-PAINT)
ImageToneTable (PHOTO-PAINT)

L

LensCreateFromMask (PHOTO-PAINT)
LensEdit (PHOTO-PAINT)
LensNew (PHOTO-PAINT)
LineTool (PHOTO-PAINT)
LocalUndo (PHOTO-PAINT)

M

MaskAffineDistort (PHOTO-PAINT)
MaskAlign (PHOTO-PAINT)
MaskBorder (PHOTO-PAINT)
MaskBrush (PHOTO-PAINT)

MaskChannelAdd (PHOTO-PAINT)
MaskChannelDelete (PHOTO-PAINT)
MaskChannelLoad (PHOTO-PAINT)
MaskChannelName (PHOTO-PAINT)
MaskChannelSave (PHOTO-PAINT)
MaskChannelToMask (PHOTO-PAINT)
MaskCreate (PHOTO-PAINT)
MaskCreateFromPath (PHOTO-PAINT)
MaskDeFloat (PHOTO-PAINT)
MaskDistort (PHOTO-PAINT)
MaskEllipse (PHOTO-PAINT)
MaskExpand (PHOTO-PAINT)
MaskFeather (PHOTO-PAINT)
MaskFlipHorizontal (PHOTO-PAINT)
MaskFlipVertical (PHOTO-PAINT)
MaskFloaterMoveTo (PHOTO-PAINT)
MaskFloaterTranslate (PHOTO-PAINT)
MaskFreehand (PHOTO-PAINT)
MaskGrow (PHOTO-PAINT)
MaskInvert (PHOTO-PAINT)
MaskLasso (PHOTO-PAINT)
MaskLoad (PHOTO-PAINT)
MaskMagicWand (PHOTO-PAINT)
MaskPaint (PHOTO-PAINT)
MaskRectangle (PHOTO-PAINT)
MaskReduce (PHOTO-PAINT)
MaskRemove (PHOTO-PAINT)
MaskRemoveHoles (PHOTO-PAINT)
MaskRotate (PHOTO-PAINT)
MaskSave (PHOTO-PAINT)
MaskScissors (PHOTO-PAINT)
MaskSelectAll (PHOTO-PAINT)
MaskSimilar (PHOTO-PAINT)
MaskSkew (PHOTO-PAINT)
MaskSmooth (PHOTO-PAINT)
MaskStretch (PHOTO-PAINT)
MaskStroke (PHOTO-PAINT)
MaskThreshold (PHOTO-PAINT)
MaskToMaskChannel (PHOTO-PAINT)
MaskTranslate (PHOTO-PAINT)
MovieBackOne (PHOTO-PAINT)
MovieCreate (PHOTO-PAINT)
MovieDeleteFrame (PHOTO-PAINT)
MovieForward (PHOTO-PAINT)
MovieForwardOne (PHOTO-PAINT)
MovieFrameDelay (PHOTO-PAINT)
MovieFrameRate (PHOTO-PAINT)
MovieGotoFrame (PHOTO-PAINT)
MovieInsertFile (PHOTO-PAINT)
MovieInsertFrame (PHOTO-PAINT)

MovieMoveFrame (PHOTO-PAINT)
MovieRewind (PHOTO-PAINT)
MovieSelectPartial (PHOTO-PAINT)

N

NibSettings (PHOTO-PAINT)

O

ObjectAddClipMask (PHOTO-PAINT)
ObjectAddClipMaskFromMask (PHOTO-PAINT)
ObjectAddClipMaskFromTransparency (PHOTO-PAINT)
ObjectAffineDistort (PHOTO-PAINT)
ObjectAlign (PHOTO-PAINT)
ObjectClip (PHOTO-PAINT)
ObjectClipToParent (PHOTO-PAINT)
ObjectColorTransparencyTool (PHOTO-PAINT)
ObjectCombine (PHOTO-PAINT)
ObjectCreate (PHOTO-PAINT)
ObjectCreateFromBackground (PHOTO-PAINT)
ObjectDefringe (PHOTO-PAINT)
ObjectDelete (PHOTO-PAINT)
ObjectDistort (PHOTO-PAINT)
ObjectDropShadow (PHOTO-PAINT)
ObjectDuplicate (PHOTO-PAINT)
ObjectEdit (PHOTO-PAINT)
ObjectEditAll (PHOTO-PAINT)
ObjectEditNone (PHOTO-PAINT)
ObjectEditSelected (PHOTO-PAINT)
ObjectEditTransparency (PHOTO-PAINT)
ObjectFeather (PHOTO-PAINT)
ObjectFlipHorizontal (PHOTO-PAINT)
ObjectFlipVertical (PHOTO-PAINT)
ObjectGroup (PHOTO-PAINT)
ObjectMerge (PHOTO-PAINT)
ObjectMergeMode (PHOTO-PAINT)
ObjectName (PHOTO-PAINT)
ObjectNew (PHOTO-PAINT)
ObjectOpacity (PHOTO-PAINT)
ObjectOrder (PHOTO-PAINT)
ObjectOrderChange (PHOTO-PAINT)
ObjectProperties (PHOTO-PAINT)
ObjectRemoveClipMask (PHOTO-PAINT)
ObjectRemoveMatte (PHOTO-PAINT)
ObjectRotate (PHOTO-PAINT)
ObjectSelect (PHOTO-PAINT)
ObjectSelectAll (PHOTO-PAINT)
ObjectSelectNone (PHOTO-PAINT)
ObjectSkew (PHOTO-PAINT)
ObjectStretch (PHOTO-PAINT)
ObjectTagWWWURL (PHOTO-PAINT)
ObjectThreshold (PHOTO-PAINT)

ObjectToggleClipMask (PHOTO-PAINT)
ObjectToggleLinkClipMask (PHOTO-PAINT)
ObjectTranslate (PHOTO-PAINT)
ObjectTransparencyTool (PHOTO-PAINT)
ObjectUngroup (PHOTO-PAINT)
ObjectURLInfo (PHOTO-PAINT)
ObjectVisible (PHOTO-PAINT)
OverprintColor (PHOTO-PAINT)

P

PaletteColor (PHOTO-PAINT)
PathCreate (PHOTO-PAINT)
PathCreateFromMask (PHOTO-PAINT)
PathDelete (PHOTO-PAINT)
PathEnd (PHOTO-PAINT)
PathImportVector (PHOTO-PAINT)
PathLoad (PHOTO-PAINT)
PathNew (PHOTO-PAINT)
PathNode (PHOTO-PAINT)
PathSave (PHOTO-PAINT)
PathStroke (PHOTO-PAINT)
PolygonTool (PHOTO-PAINT)
PressureSettings (PHOTO-PAINT)

R

RandomSeed (PHOTO-PAINT)
Rectangle (PHOTO-PAINT)
RectangleTool (PHOTO-PAINT)
RepeatSettings (PHOTO-PAINT)

S

SelectionMoveTo (PHOTO-PAINT)
SetDocumentInfo (PHOTO-PAINT)
SetDocVisible (PHOTO-PAINT)
SetPaintColor (PHOTO-PAINT)
SetPaperColor (PHOTO-PAINT)
SetVisible (PHOTO-PAINT)
StartCloneDraw (PHOTO-PAINT)
StartDraw (PHOTO-PAINT)
SymmetrySettings (PHOTO-PAINT)

T

TextSettings (PHOTO-PAINT)
TextTool (PHOTO-PAINT)
ToleranceSettings (PHOTO-PAINT)
TransparencyBrushTool (PHOTO-PAINT)

File commands (PHOTO-PAINT)

BindToActiveDocument (PHOTO-PAINT)

.BindToActiveDocument

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

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
```

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

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FileAcquireWithFile (PHOTO-PAINT)

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

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.

Parameter	Description
.FileName	Lets you specify the name of the file.
.Left	Lets you specify the X-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the image's bounding box 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 "TEST.1.CPT", 0, 0, 0, 0, 0
```

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

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FileClose (PHOTO-PAINT)

.FileClose

This command closes the active image.

Example

```
.FileClose
```

This example closes the active image.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FileNew (PHOTO-PAINT)

.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,* .Backgroundless = *boolean*

This command defines the characteristics of a new image.

Parameter	Description
.Width	Lets you specify the width of the new image in pixels.
.Height	Lets you specify the height of the new image in pixels.
.Type	Indicates the image type: 1 = RGB 2 = 256 Grayscale 3 = Black and white 4 = Paletted 5 = CMYK 7 = LAB
.HRes	Lets you specify the horizontal resolution of the image in dots per inch (dpi).
.VRes	Lets you specify 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 .PartialFile option is disabled.
.NumberFrames	If the movie file option is enabled (i.e. set to TRUE) enter the number of frames you would like the movie file to contain.
.Left	Lets you specify the X-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin. (Applies only if the new image is a partial file.).
.Top	Lets you specify the Y-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin. (Applies only if the new image is a partial file.).
.Right	Lets you specify the X-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin. (Applies only if the new image is a partial file.).
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin. (Applies only if the new image is a partial file.).
.Color1	Lets you specify the first color component for .Type. For example, Red is the first color component for RGB. Use RGB for 16 Color and 256 Color images.
.Color2	Lets you specify the second color component for .Type. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0. Use RGB for 16 Color and 256 Color images.
.Color3	Lets you specify the third color component for .Type. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0. Use RGB for 16 Color and 256 Color images.
.Color4	Lets you specify the fourth color component for .Type. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0. Use RGB for 16 Color and 256 Color images.
.Backgroundless	Set to TRUE (-1) to create an image without a background. Set to FALSE (0) to create an image with a background.

Example

```
.FileNew 500, 500, 3, 100, 100, FALSE, FALSE, 1, 0, 0, 0, 0, 0, 0, 0, 0, FALSE
```

This example creates a new file that is 500 pixels by 500 pixels in size and has a black background.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FileOpen (PHOTO-PAINT)

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

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

Parameter	Description
.FileName	Lets you specify the path and name of the file to be opened.
.Left	Lets you specify the X-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the image's bounding box 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 4 = Crop
.StartFrame	Lets you specify the starting index of the frame range to load from a movie file.
.EndFrame	Lets you specify the ending index of the frame range to load from a movie file.

Example

```
.FileOpen "TEST.1.CPT", 0, 0, 0, 0, 0, 1, 1
```

This example opens the Corel PHOTO-PAINT file named TEST.1.CPT.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

FilePrint (PHOTO-PAINT)

.FilePrint

This command sends the current document to the printer.

Example

```
.FilePrint
```

This example sends the active document to the printer.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FileRevert (PHOTO-PAINT)

.FileRevert

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

Example

```
.FileRevert
```

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

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FileSave (PHOTO-PAINT)

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

This command saves the current image.

Parameter	Description
.FileName	Lets you specify the path and name of the file to be saved.
.FilterID	Lets you specify 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 Bitmap (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) 789 = Picture Publisher 4 (PB4) 790 = MACPaint Bitmap (MAC) 792 = OS/2 Bitmap (BMP) 793 = Wavelet Compressed Bitmap (WI) 800 = CALS Compressed Bitmap (CAL) 802 = Portable Network Graphic (PNG) 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 (CPT) 1793 = Corel CMX 6.0 1794 = Corel CMX 5.0 1795 = CorelDRAW (CDR)
.Compression	Lets you specify the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.FileSave "TEST1.BMP", 769, 0
```

This example saves the file named TEST1.BMP in Windows Bitmap format with no compression applied.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

FileSelectPartialArea (PHOTO-PAINT)

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

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

Parameter	Description
.Left	Lets you specify the X-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.

 **Note**

The image 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
```

This example opens the specified area of the image.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FilterGIF (PHOTO-PAINT)

.FilterGIF .InvertMask = *boolean*, .Interlace = *boolean*, .Transparent = *long*, .Index = *long*, .Delay = *long*, .Red = *long*, .Green = *long*, .Blue=*long*

This command sets the CompuServe Bitmap (GIF) filter information for the FileSave command.

Parameter	Description
.InvertMask	Set to TRUE (-1) to invert the transparency mask.
.Interlace	Set to TRUE (-1) for an interlaced GIF.
.Transparent	Lets you specify the transparency method: 0 = None 1 = Sets the color specified by the .Index parameter to be transparent 2 = Uses a transparency mask
.Index	Lets you specify the color in the current palette to make transparent. Valid values range from 0 to 255.
.Delay	Lets you specify the delay, in hundredths of a second, between successive frames of an animated GIF.
.Red	Lets you specify the red channel of the area outside the transparency mask. Valid values range from 0 to 255.
.Green	Lets you specify the green channel of the area outside the transparency mask. Valid values range from 0 to 255.
.Blue	Lets you specify the blue channel of the area outside the transparency mask. Valid values range from 0 to 255.

Example

```
.FilterGIF 0, 1, 1, 200, 0, 0, 0, 0
.FileSave "test.gif", 773, 0
```

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

FilterJPG (PHOTO-PAINT)

.FilterJPG .Compression= *long*, .Progressive = *long,* .Smoothing = *long,* .SubFormat = *long,* .Optimized = *long*

This command sets the JPEG filter information for the FileSave command.

Parameter	Description
.Compression	Lets you specify the compression factor of the bitmap. Valid values range from 2 (large, high-quality file) to 255 (small, low-quality file).
.Progressive	Set to TRUE (-1) to create a progressive jpeg.
.Smoothing	Lets you specify a smoothing value for the image.
.SubFormat	Lets you specify an encoding method sub format. You can select Standard or Optional compression formats.
.Optimized	Lets you specify an optimal encoding method.

Example

```
.FilterJPG 51, TRUE, 46, 0 FALSE
```

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FilterOS2 (PHOTO-PAINT)

.FilterOS2 .Format = *long*

This command sets the OS/2 Bitmap (BMP) filter information for the FileSave command.

Parameter	Description
.Format	Sets the export format: 0 = Standard format (OS/2 version 1.3) 1 = Enhanced format (OS/2 version 2.0 or later)

Example

```
.FilterOS2 1
.FileSave "test.bmp", 792, 0
```

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FilterPNG (PHOTO-PAINT)

.FilterPNG .Interlace = *long*

This command sets the Portable Network Graphic (PNG) filter information for the FileSave command.

Parameter	Description
.Interlace	Set to TRUE (-1) to create an interlaced PNG.

Example

```
.FilterPNG 1
.FileSave "Test.png", 802, 0
```

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FilterTGA (PHOTO-PAINT)

.FilterTGA .Format = *long*

This command sets the Targa Bitmap (TGA) filter information for the FileSave command.

Parameter	Description
.Format	Lets you specify the bitmap format: 0 = Normal 1 = Enhanced

Example

```
.FilterTGA 1
.FileSave "test.tga", 771, 0
```

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FilterWVL (PHOTO-PAINT)

.FilterWVL .Compression = *long,* .Speed = *long,* .Path = *long,* .Contrast = *long,* .Edge = *long*

This command sets the Wavelet Compressed Bitmap (WI) filter information for the FileSave command.

Parameter	Description
.Compression	Lets you specify the compression factor of the bitmap. Valid values range from 1 (large, high-quality file) to 100 (small, low-quality file).
.Speed	Lets you specify the speed of the encoding method 0=Normal 1=Fast
.Path	Lets you specify the encoding path 0=One 1=Two 2=Three
.Contrast	Sets the level of contrast. Valid values range from 0 to 8.
.Edge	Sets the edging of the image. Valid values range from 0 to 8.

Example

```
.FilterWVL 41, 0, 0, 0, 0
```

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

SetDocumentInfo (PHOTO-PAINT)

.SetDocumentInfo .Width = *long*, .Height = *long*

This command defines an image size to use as a reference for performing the operations in the script. If you define the image size using SetDocumentInfo, you can run your script on images of any size and the script will scale its operations to match.

Parameter	Description
.Width	Lets you specify the reference image width.
.Height	Lets you specify the reference image height.

Example

```
.SetDocumentInfo 640, 480
.MaskRectangle 20, 20, 620, 460, 0, 0
```

This example can create a MaskRectangle on any image size with the same relative size as the original.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

SetDocVisible (PHOTO-PAINT)

.SetDocVisible .Show = *boolean*

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

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

Example

```
.SetDocVisible -1
```

This example displays the active Corel PHOTO-PAINT document.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

SetVisible (PHOTO-PAINT)

.SetVisible .Show = *boolean*

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

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

Example

```
.SetVisible -1
```

This example shows the Corel PHOTO-PAINT application.

{button ,AL(`OVR1 File commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Edit commands (PHOTO-PAINT)

EditCheckpoint (PHOTO-PAINT)

.EditCheckpoint

This command saves a copy of the image to a temporary file. Additions or edits to the image that are performed after 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.

Note

Issuing this command clears the undo list.

Example

```
.EditCheckpoint
```

This example saves the image at its current state.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditClear (PHOTO-PAINT)

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

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 is ignored.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EditClear 5, 255, 255, 0, 0
```

This example clears the current document and removes any editing that has been performed.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditClearClipboard (PHOTO-PAINT)

.EditClearClipboard

This command clears the Clipboard of all information, which conserves memory and reduces execution time.

Example

```
.EditClearClipboard
```

This example removes all items previously placed on the Clipboard.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditCopy (PHOTO-PAINT)

.EditCopy

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

Example

```
.EditCopy
```

This example copies the selected object to the Clipboard.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditCopyToFile (PHOTO-PAINT)

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

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

Parameter	Description
.FileName	The path and file name of the destination file.
.FilterID	Lets you specify 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 Bitmap (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	Lets you specify the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.EditCopyToFile "TEST1.CPT", 1792, 0
```

This example copies the selected object to the PHOTO-PAINT file named TEST1.CPT.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EditCopyVisible (PHOTO-PAINT)

.EditCopyVisible

This command allows you to copy all visible objects, masked selections, or floating selections in the active image to the Clipboard.

Example

```
.EditCopyVisible
```

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditCut (PHOTO-PAINT)

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

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

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Note

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

Example

```
.EditCut 5, 255, 255, 0, 0
```

This example removes the selected object from the document and places it on the Clipboard.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditCutIntoSelection (PHOTO-PAINT)

.EditCutIntoSelection

This command cuts a selection from inside another selection.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EditCutMask (PHOTO-PAINT)

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

This command cuts the area inside a mask and copies it to the Clipboard. The space behind the cut section reverts to the background color.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EditCutMask 5, 255, 255, 0, 0
```

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditCutSelection (PHOTO-PAINT)

.EditCutSelection

This command cuts a floating selection onto the Clipboard. Unlike the EditCutMask command, the area behind the selection is unchanged.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditFill (PHOTO-PAINT)

.EditFill .MergeMode = *long*, .StartTransparency = *long*, .EndTransparency = *long*, .GradientType = *long*, .Handles = *long*, .X1 = *long*, .Y1 = *long*, .X2 = *long*, .Y2 = *long*, .X3 = *long*, .Y3 = *long*

This command applies a fill to the current image using settings specified in a block of commands. EditFill command blocks must end with an EndEditFill command.

Parameter	Description
.MergeMode	Lets you specify 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
.StartTransparency	Lets you specify the starting transparency value for a gradient fill.
.EndTransparency	Lets you specify the ending transparency value for a gradient fill.
.GradientType	Lets you specify the gradient type: 0 = None 1 = Flat 2 = Linear 3 = Elliptical 4 = Radial 5 = Rectangular 6 = Square 7 = Conical
.Handles	Lets you specify the number of handles used to define the gradient. Valid values range from 1 to 3, depending on the type of gradient.
.X1	Lets you specify the horizontal coordinate of the first gradient handle.
.Y1	Lets you specify the vertical coordinate of the first gradient handle.
.X2	Lets you specify the horizontal coordinate of the second gradient handle.
.Y2	Lets you specify the vertical coordinate of the second gradient handle.
.X3	Lets you specify the horizontal coordinate of the third gradient handle.
.Y3	Lets you specify the vertical coordinate of the third gradient handle.

Example

```
.EditFill 0, 0, 100, 3, 3, 294, 240, 79, 164, 114, 61
    .FillFountainColor 5, 2, 74, 123, 0, 0, 0
    .FillFountainColor 5, 255, 255, 255, 0, 50, 1
    .FillFountainColor 5, 63, 125, 122, 0, 100, 2
    .FillFountainApply 3, 3, 256, 135, 0, 0, 0, 50
    .EndEditFill
```

This example creates an Elliptical Fountain Fill with a starting transparency of 0 and an ending transparency of 100.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditPasteDocument (PHOTO-PAINT)

.EditPasteDocument

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

Example

```
.EditPasteDocument
```

This example inserts a copy of the object most recently placed on the Clipboard into a new document.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditPasteFromFile (PHOTO-PAINT)

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

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

Parameter	Description
.FileName	Lets you specify the path and name of the source file.
.Left	Lets you specify the X-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.
.LoadType	Lets you specify the image loading method: 0 = All 2 = Resample 4 = Crop
.ptX	Lets you specify the X-coordinate in the active image to place the center of the pasted image.
.ptY	Lets you specify the Y-coordinate in the active image to place the center of the pasted image.

Example

```
.EditPasteFromFile "CLAMSHEL.BMP", 0, 0, 0, 0, 0, 180, 255
```

This example pastes the bitmap file named CLAMSHEL.BMP into the open image.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditPasteIntoSelection (PHOTO-PAINT)

.EditPasteIntoSelection

This command pastes the Clipboard contents into the current selection.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EditPasteObject (PHOTO-PAINT)

.EditPasteObject .ptX = *long*, .ptY = *long*, .FileName = *string*

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

Parameter	Description
.ptX	Lets you specify the X-coordinate in the active image to place the center of the pasted image.
.ptY	Lets you specify the Y-coordinate in the active image to place the center of the pasted image.
.FileName	If filename is "" (Null) then the image is pasted from the Clipboard, instead of from the "FileName".

Example

```
.EditPasteObject 160, 120, ""
```

This example pastes a copy of the object most recently placed in the Clipboard into the current document.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditPasteSelection (PHOTO-PAINT)

.EditPasteSelection .ptX = *long*, .ptY = *long*, .FileName = *string*

This command pastes the contents of the Clipboard into the active image as a selection.

Parameter	Description
.ptX	Lets you specify the X-coordinate in the active image to place the center of the pasted image.
.ptY	Lets you specify the Y-coordinate in the active image to place the center of the pasted image.
.FileName	If filename is "" (Null) then the image is pasted from the Clipboard, instead of from the "FileName".

Example

```
.EditPasteSelection 100, 200, ""
```

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EditRestoreCheckpoint (PHOTO-PAINT)

.EditRestoreCheckpoint

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

Example

```
.EditRestoreCheckpoint
```

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

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EditSingleObject (PHOTO-PAINT)

.EditSingleObject .Mode = *long*

This command switches between the three editing modes.

Parameter	Description
.Mode	Lets you specify the editing mode: 0 = Multiple object 1 = Single object 3 = Layer object

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndEditFill (PHOTO-PAINT)

.EndEditFill

This command ends an EditFill command block.

Example

```
.EditFill 0, 0, 100, 3, 3, 294, 240, 79, 164, 114, 61
    .FillFountainColor 5, 2, 74, 123, 0, 0, 0
    .FillFountainColor 5, 255, 255, 255, 0, 50, 1
    .FillFountainColor 5, 63, 125, 122, 0, 100, 2
    .FillFountainApply 3, 3, 256, 135, 0, 0, 0, 50
    .EndEditFill
```

This example creates an Elliptical Fountain Fill with a starting transparency of 0 and an ending transparency of 100.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FadeLastCommand (PHOTO-PAINT)

.FadeLastCommand .Percent = *long,* .MergeMode = long

This command reduces the effect of the most recent operation gradually.

Parameter	Description
.Percent	Lets you specify the amount by which the operation is faded. Values range from 0 to 100. If you set the percent to zero, the last command is not faded. If you set the percent to 100, the last command is made transparent.
.MergeMode	Lets you specify 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

Example

```
.FadeLastCommand 77, 0
```

This example fades the last recorded command by 77% in Normal mode.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

LocalUndo (PHOTO-PAINT)

.LocalUndo .Width = *long*, .Flatten = *long*, .Rotate = *long*, .NibShape = *long*, .Transparency = *long*, .SoftEdge = *long*

This command reverts an area described by a series of Draw commands to the state it was in before the last operation. A LocalUndo command block must contain a series of StartDraw and ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 100%.
.Rotate	Lets you specify the rotation of the nib. Valid values range from 0 to 360 degrees.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Transparency	Lets you specify the transparency of the nib. Valid values range from 0 to 100%.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.LocalUndo 20, 100, 0, 0, 0, 0
    .StartDraw 40320, 8832, 0, 0
    .ContinueDraw 39040, 8832, 0, 0
    ...
    .ContinueDraw 59678, 18166, 0, 0
    .EndDraw
```

This example replaces the portion of your image defined by the Draw commands with the defined paper color.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

SelectionMoveTo (PHOTO-PAINT)

.SelectionMoveTo .Left = *long*, .Bottom = *long*

This command moves the current selection to the given coordinates.

Parameter	Description
.Left	Lets you specify the horizontal displacement of the bottom-left corner of the selection, in pixels. The displacement is relative to the original position of the selection, rather than to the origin.
.Bottom	Lets you specify the vertical displacement of the bottom-left corner of the selection, in pixels. The displacement is relative to the original position of the selection, rather than to the origin.

{button ,AL(`OVR1 Edit commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Get commands (PHOTO-PAINT)

GetChannelCount (PHOTO-PAINT)

ReturnValue& = .GetChannelCount()

This function returns a value indicating the number of mask channels saved.

Return Value

The number of mask channels saved.

Example

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

This example displays the number of mask channels. The variable cnt is assigned a value of 2.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetChannelName (PHOTO-PAINT)

ReturnValue& = .GetChannelName(long ChannelID)

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

Return Value

Returns a string containing the name of the specified channel.

Parameter	Description
.ChannelID	Lets you specify the channel. Channels for the active document are listed in the Channels Roll-Up. The channels are numbered and identified according to their position in the list ▸ the first listed channel is identified as 0, the second listed channel is identified as 1, and so on.

Example

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

This example sets and then returns the name of the second channel in the channel list.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetCurrentMovieFrame (PHOTO-PAINT)

ReturnValue& = .GetCurrentMovieFrame()

This function returns the number of frames in the current movie.

Return Value

Returns a value indicating the number of frames in the current movie.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentCount (PHOTO-PAINT)

ReturnValue& = .GetDocumentCount()

This function returns the number of open documents

Return Value

Returns a value indicating the number of open documents.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentHeight (PHOTO-PAINT)

ReturnValue& = .GetDocumentHeight()

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

Return Value

Indicates the height of the active image.

Example

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

This example returns the height of the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentModified (PHOTO-PAINT)

ReturnValue& = .GetDocumentModified()

This command specifies whether the document has been modified since it was loaded.

Return Value

The .GetDocumentModified function returns one of the following values:

- TRUE (-1) ▶ the document has been modified since it was loaded
- FALSE (0) ▶ the document has not been modified since it was loaded

---

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetDocumentIsMovie (PHOTO-PAINT)

ReturnValue = .GetDocumentIsMovie()

This function returns a value indicating whether the active document is a movie file.

Return Value

The .GetDocumentIsMovie function returns one of the following values:

- TRUE (-1)▸Document is a movie file
- FALSE (0)▸Document is not a movie file.

Example

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

This example determines whether the document is a movie file.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentIsPartial (PHOTO-PAINT)

ReturnValue = .GetDocumentIsPartial()

This function returns a value indicating whether the active document is a partial file

Return Value

The .GetDocumentIsPartial returns one of the following value:

- TRUE (-1)▸Document is a partial file
- FALSE (0)▸Document is not a partial file.

Example

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

This example determines whether the document is a partial file.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentName (PHOTO-PAINT)

ReturnValue& = .GetDocumentName()

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

Return Value

Returns a string containing the name and extension of the active image.

Example

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

This example determines the name of the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentType (PHOTO-PAINT)

ReturnValue& = .GetDocumentType()

This function returns a value indicating the PHOTO-PAINT image type of the active document:

Return Value

The .GetDocumentType function returns one of the following values:

- 1 RGB
- 2 256 Grayscale
- 3 Black and white
- 4 Paletted
- 5 CMYK
- 6 Duotone
- 7 LAB

Example

```
docType& = .GetDocumentType()
```

This example determines the paint image type of the active document.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetDocumentWidth (PHOTO-PAINT)

ReturnValue& = .GetDocumentWidth()

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

Return Value

Indicates the width of the active image.

Example

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

This example returns the width of the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentXdpi (PHOTO-PAINT)

ReturnValue& = .GetDocumentXdpi()

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

Return Value

Indicates the horizontal resolution of the active image.

Example

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

This example returns the horizontal resolution of the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetDocumentYdpi (PHOTO-PAINT)

ReturnValue& = .GetDocumentYdpi()

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

Return Value

Indicates the vertical resolution of the active image.

Example

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

This example returns the vertical resolution of the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetFillColor (PHOTO-PAINT)

ReturnValue = .GetFillColor(.ColorModel = *long**, .Color1 = *long**, .Color2 = *long**, .Color3 = *long**, long* Color4)

This function returns the fill color of the selected object. The number and type of .Color variables returned depends on the current Color Model. The function will return FALSE values (0) if the selected object is filled with anything other than a solid color.

Parameter	Description
.ColorModel	Returns the currently active Color Model: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Returns the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Returns the second color component for .ColorModel. For example, Green is the second color component for RGB.
.Color3	Returns the third color component for .ColorModel. For example, Blue is the third color component for RGB.
.Color4	Returns the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetFrameCount (PHOTO-PAINT)

ReturnValue& = .GetFrameCount()

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

Example

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

This example determines the frame count of the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetMaskPresent (PHOTO-PAINT)

ReturnValue& = .GetMaskPresent()

This function returns a value indicating whether a mask is present.

Return Value

The .GetMaskPresent function returns one of the following values:

- 1 Mask is present
- 0 Mask is not present

Example

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

This example determines whether a mask is present.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetMaskRectangle (PHOTO-PAINT)

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

This function determines the coordinates of the active mask.

Parameter	Description
.Left	Returns the X-coordinate of the upper-left corner of the mask's bounding box in pixels, relative to the origin.
.Top	Returns the Y-coordinate of the upper-left corner of the mask's bounding box in pixels, relative to the origin.
.Right	Returns the X-coordinate of the lower-right corner of the mask's bounding box in pixels, relative to the origin.
.Bottom	Returns the Y-coordinate of the lower-right corner of the mask's bounding box in pixels, relative to the origin.

Example

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

This example returns the coordinates of the mask's bounding box.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetObjectCount (PHOTO-PAINT)

ReturnValue& = .GetObjectCount()

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

Note

The background is not included in the object count.

Example

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

This example determines the number of objects in the active image.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetObjectIsEditable (PHOTO-PAINT)

ReturnValue = .GetObjectIsEditable(long ObjectID)

This function returns a value indicating whether the specified object is editable.

Return Value

The .GetObjectIsEditable function returns one of the following values:

- TRUE (-1) ▸ editable
- FALSE (0) ▸ not editable

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

Example

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

This example determines whether the third object is editable.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetObjectIsSelected (PHOTO-PAINT)

ReturnValue = .GetObjectIsSelected(long ObjectID)

This function returns a value indicating whether the specified object is selected.

Return Value

The .GetObjectIsSelected function returns one of the following value:

- TRUE (-1)▸Selected
- FALSE (0)▸Not selected

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

▸ **Note**

Since the background cannot be selected, .ObjectID must be greater than 0.

Example

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

This example determines whether the third object is selected.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetObjectIsVisible (PHOTO-PAINT)

ReturnValue = .GetObjectIsVisible(long ObjectID)

This function returns a value indicating whether the specified object is visible or hidden.

Return Value

The .GetObjectIsVisible function returns one of the following values:

- TRUE (-1) = Visible
- FALSE (0) = Hidden

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

Example

```
status& = .GetObjectIsVisible(4)
MESSAGE status&
```

This example determines whether the fifth object is visible.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetObjectMergeMode (PHOTO-PAINT)

ReturnValue& = .GetObjectMergeMode(long ObjectID)

This function returns a value indicating the Merge Mode for the specified object:

Return Value

The .GetObjectMergeMode function returns one of the following values:

- 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

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

Note

Since the background merge mode cannot be queried, .ObjectID must be greater than 0.

Example

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

This example determines the merge mode of the third object.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetObjectName (PHOTO-PAINT)

ReturnValue& = .GetObjectName(long ObjectID)

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

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

Example

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

This example determines the name of the third object.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetObjectOpacity (PHOTO-PAINT)

ReturnValue& = .GetObjectOpacity(long ObjectID)

This function returns a value (1 to 100) indicating the opacity of the specified object.

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

Note

Since the background merge mode cannot be queried, .ObjectID must be greater than 0.

Example

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

This example determines the opacity of the third object.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetObjectProperties (PHOTO-PAINT)

.GetObjectProperties .ObjectID = *long*, .Name = *string*, .Opacity = *long**, .MergeMode = *long**, .Visible = *boolean**, .Clipped = *boolean**, .Enabled = *boolean**, .Linked = *boolean**

This function returns the attributes or properties of the specified object.

Parameter	Description
.ObjectID	Specifes the index number of the object.
.Name	Lets you specify the name of the object.
.Opacity	Lets you specify the opacity of the object.*
.MergeMode	.MergeMode 0 = Normal 1 = Add 2 = Subtract 3 = Difference 4 = Multiply 5 = Divide 6 = Lighter 7 = Darker 8 = Texturize 9 = Color 10 = Hue 11 = Saturation Lets you specify the Merge Mode: 12 = Lum 13 = Invert 14 = And 15 = Or 16 = Xor 17 = Red 18 = Green 19 = Blue 20 = Cyan 21 = Magenta 22 = Yellow 23 = Black
.Visible	Lets you specify whether the object is visible or hidden. If the parameter is set to TRUE (-1), the object is visible. If the parameter is set to FALSE (0), the object is invisible.
.Clip	Lets you specify whether the object is clipped to its parent object. If the parameter is set to TRUE (-1), the object is clipped to its parent object. Otherwise, the parameter is set to FALSE (0).
.Enable	Lets you specify whether the object's clip mask is enabled. If the parameter is set to TRUE (-1), the object's clip mask is enabled. If the parameter is set to FALSE (0), the object's clip mask is disabled.
.Link	Lets you specify whether the object is linked to its clip mask. If the parameter is set to TRUE (-1), the object is linked to its clip mask. Otherwise, the parameter is set to FALSE (0).

Example

```
.GetObjectProperties
```

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetObjectRectangle (PHOTO-PAINT)

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

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

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.
.Left	Returns the X-coordinate of the upper-left corner of the object's bounding box in pixels, relative to the origin.
.Top	Returns the Y-coordinate of the upper-left corner of the object's bounding box in pixels, relative to the origin.
.Right	Returns the X-coordinate of the lower-right corner of the object's bounding box in pixels, relative to the origin.
.Bottom	Returns the Y-coordinate of the lower-right corner of the object's bounding box in pixels, relative to the origin.

Example

```
.GetObjectRectangle 5, l&, t&, r&, b&
MESSAGE l&
MESSAGE t&
MESSAGE r&
MESSAGE b&
```

This example determines the coordinates of the sixth object's bounding box.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetPaintColor (PHOTO-PAINT)

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

This function returns the color that is assigned to the Brush tool. The number and type of .Color variables returned depends on the current Color Model.

Parameter	Description
.ColorModel	Returns the currently active Color Model: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Returns the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Returns the second color component for .ColorModel. For example, Green is the second color component for RGB.
.Color3	Returns the third color component for .ColorModel. For example, Blue is the third color component for RGB.
.Color4	Returns the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetPaintVersion (PHOTO-PAINT)

ReturnValue& = .GetPaintVersion()

This function returns a string containing the version information. The string includes the word "Version" followed by the PHOTO-PAINT version number. For example, "Version 6.00.118".

Example

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

This example returns the version information.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetPaperColor (PHOTO-PAINT)

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

This function returns the current paper (background) color. The number and type of .Color variables returned depends on the current Color Model.

Parameter	Description
.ColorModel	Returns the currently active Color Model: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Returns the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Returns the second color component for .ColorModel. For example, Green is the second color component for RGB.
.Color3	Returns the third color component for .ColorModel. For example, Blue is the third color component for RGB.
.Color4	Returns the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetPartialDocumentHeight (PHOTO-PAINT)

ReturnValue& = .GetPartialDocumentHeight()

This function returns the height of that portion of an image you choose to load using the Partial Load option in the Open dialog box, or the FileOpen command. Use the GetPartialDocumentWidth function to get the width of a partial image.

Return Value

Indicates the height of the current partial image, in pixels.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetPartialDocumentWidth (PHOTO-PAINT)

ReturnValue& = .GetPartialDocumentWidth()

This function returns the width of that portion of an image you choose to load using the Partial Load option in the Open dialog box, or the FileOpen command. Use the GetPartialDocumentHeight function to get the height of a partial image.

Return Value

The width of the current partial image, in pixels.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GetPhotoPaintDir (PHOTO-PAINT)

ReturnString$ = .GetPhotoPaintDir()

This function returns the name of the folder where Corel PHOTO-PAINT is installed.

Return Value

Returns the name of the folder where Corel PHOTO-PAINT is installed.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GetSelectedObjectsRectangle (PHOTO-PAINT)

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

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

Parameter	Description
.Left	Returns the X-coordinate of the upper-left corner of the object's bounding box in pixels, relative to the origin.
.Top	Returns the Y-coordinate of the upper-left corner of the object's bounding box in pixels, relative to the origin.
.Right	Returns the X-coordinate of the lower-right corner of the object's bounding box in pixels, relative to the origin.
.Bottom	Returns the Y-coordinate of the lower-right corner of the object's bounding box 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&
```

This example returns the coordinates of the selected object's bounding box.

{button ,AL(`OVR1 Get commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Image commands (PHOTO-PAINT)

DuotoneHandle (PHOTO-PAINT)

.DuotoneHandle .ToneNumber = *long*, .HandleNumber = *long*, .X = *long*, .Y = *long*

This command specifies the coordinates of the control handles on the tone curve of an ImageConvertDuotone command. This command must appear in an ImageConvertDuotone command block.

Parameter	Description
.ToneNumber	Lets you specify the tone to change. Valid values range from 0 to 3.
.HandleNumber	Lets you specify the tone curve handle you want to change.
.X	Lets you specify the horizontal coordinate of the tone curve handle. Valid values range from 0 to 255.
.Y	Lets you specify the vertical coordinate of the tone curve handle. Valid values range from 0 to 100.

Example

```
.ImageConvertDuotone 2, TRUE
    .OverprintColor 0, 2, 26, 97, 0
    .OverprintColor 1, 0, 100, 0, 100
    .OverprintColor 2, 100, 0, 0, 100
    ...
    .OverprintColor 10, 100, 100, 100, 100

    .DuotoneInfo 0, 3, 0, 0, 0, 255
    .DuotoneHandle 0, 0, 0, 0
    .DuotoneHandle 0, 1, 117, 66
    .DuotoneHandle 0, 2, 255, 100

    .DuotoneInfo 1, 4, 0, 0, 255, 0
    .DuotoneHandle 1, 0, 0, 0
    .DuotoneHandle 1, 1, 85, 73
    .DuotoneHandle 1, 2, 166, 40
    .DuotoneHandle 1, 3, 255, 100
    .EndConvertDuotone
```

This example converts a grayscale image to a duotone with black and yellow channels.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

DuotoneInfo (PHOTO-PAINT)

.DuotoneInfo .ToneNumber = *long*, .Handles = *long*, .Cyan = *long*, .Magenta = *long*, .Yellow = *long*, .Black = *long*

This command sets the attributes of the individual tones for the ImageConvertDuotone command. This command must appear in an ImageConvertDuotone command block.

Parameter	Description
.ToneNumber	Lets you specify the tone to change. Valid values range from 0 to 3.
.Handles	Lets you specify the number of handles in the tone curve.
.Cyan	Lets you specify the Cyan channel of the tone.
.Magenta	Lets you specify the Magenta channel of the tone.
.Yellow	Lets you specify the Yellow channel of the tone.
.Black	Lets you specify the Black channel of the tone.

Example

```
.ImageConvertDuotone 2, TRUE
    .OverprintColor 0, 2, 26, 97, 0
    .OverprintColor 1, 0, 100, 0, 100
    .OverprintColor 2, 100, 0, 0, 100
    ...
    .OverprintColor 10, 100, 100, 100, 100

    .DuotoneInfo 0, 3, 0, 0, 0, 255
    .DuotoneHandle 0, 0, 0, 0
    .DuotoneHandle 0, 1, 117, 66
    .DuotoneHandle 0, 2, 255, 100

    .DuotoneInfo 1, 4, 0, 0, 255, 0
    .DuotoneHandle 1, 0, 0, 0
    .DuotoneHandle 1, 1, 85, 73
    .DuotoneHandle 1, 2, 166, 40
    .DuotoneHandle 1, 3, 255, 100
    .EndConvertDuotone
```

This example converts a grayscale image to a duotone with black and yellow channels.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndColorTable (PHOTO-PAINT)

.EndColorTable

This command ends an ImageColorTable command block.

Example

```
.ImageColorTable 256
    .PaletteColor 5, 255, 0, 255, 0, 0
    .PaletteColor 5, 249, 0, 255, 0, 1
    ...
    .PaletteColor 5, 255, 255, 255, 0, 255
    .EndColorTable
```

This example changes the existing palette to a new palette containing 256 colors. A PaletteColor command is needed for each color in the new palette.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndConvertDuotone (PHOTO-PAINT)

.EndConvertDuotone

This command ends an ImageConvertDuotone command block.

Example

```
.ImageConvertDuotone 2, TRUE
    .OverprintColor 0, 2, 26, 97, 0
    .OverprintColor 1, 0, 100, 0, 100
    .OverprintColor 2, 100, 0, 0, 100
    ...
    .OverprintColor 10, 100, 100, 100, 100

    .DuotoneInfo 0, 3, 0, 0, 0, 255
    .DuotoneHandle 0, 0, 0, 0
    .DuotoneHandle 0, 1, 117, 66
    .DuotoneHandle 0, 2, 255, 100

    .DuotoneInfo 1, 4, 0, 0, 255, 0
    .DuotoneHandle 1, 0, 0, 0
    .DuotoneHandle 1, 1, 85, 73
    .DuotoneHandle 1, 2, 166, 40
    .DuotoneHandle 1, 3, 255, 100
    .EndConvertDuotone
```

This example converts a grayscale image to a duotone with black and yellow channels.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndConvertPaletted (PHOTO-PAINT)

.EndConvertPaletted

This command ends an ImageConvertPaletted command block.

Example

```
.ImageConvertPaletted 1, 216
    .PaletteColor 5, 0, 0, 0, 0, 0
    .PaletteColor 5, 51, 0, 0, 0, 1
    .PaletteColor 5, 102, 0, 0, 0, 2
    .PaletteColor 5, 153, 0, 0, 0, 3
    ...
    .PaletteColor 5, 153, 255, 255, 0, 213
    .PaletteColor 5, 204, 255, 255, 0, 214
    .PaletteColor 5, 255, 255, 255, 0, 215
    .EndConvertPaletted
```

This example converts an image to a paletted image and maps the existing colors to the new palette colors using a Uniform render.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndImageEqualize (PHOTO-PAINT)

.EndImageEqualize

This command ends an ImageEqualize command block.

Example

```
.ImageEqualize 0, 5, 5, FALSE
    .ImageEqualizeChannel 0, 27, 255, 8, 249, 208
    .ImageEqualizeChannel 1, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 2, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 3, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 4, 0, 255, 0, 255, 100
    .EndImageEqualize
```

This example equalizes the overall RGB channel without enhancing any of the channels individually.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndImageSTBalance (PHOTO-PAINT)

.EndImageSTBalance

This command ends a ImageSTBalance command block.

Example

```
.ImageSTBalance 0, TRUE, TRUE, TRUE, TRUE
    ' Source low color
    .ImageSTColor 0, 5, 249, 249, 153, 0
    ' target low color
    .ImageSTColor 1, 5, 249, 249, 153, 0
    ' source mid color
    .ImageSTColor 2, 5, 249, 249, 222, 0
    ' target mid color
    .ImageSTColor 3, 5, 249, 249, 153, 0
    ' source high color
    .ImageSTColor 4, 5, 222, 189, 222, 0
    ' target high color
    .ImageSTColor 5, 5, 222, 189, 222, 0
    .EndImageSTBalance
```

This example maps each of the low, mid, and high colors to the designated target colors.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndImageToneCurve (PHOTO-PAINT)

.EndImageToneCurve

This command ends a ImageToneCurve command block.

Example

```
.ImageToneCurve
    .ImageToneTable 0, 0, 0, 0, 0
    .ImageToneTable 1, 1, 0, 1, 0
    .ImageToneTable 2, 1, 0, 1, 1
    .ImageToneTable 3, 2, 0, 2, 1
    ...
    .ImageToneTable 254, 255, 252, 255, 253
    .ImageToneTable 255, 255, 255, 255, 255
    .EndImageToneCurve
```

This example defines a new tone curve for the current image. Each of the 256 ImageToneTable commands defines one tone in the new tone curve.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageAutoEqualize (PHOTO-PAINT)

.ImageAutoEqualize .AutoBlack = *long*, .AutoWhite = *long*

This command performs a flat equalization on your image by automatically redistributing the significant pixel values of your image through the tonal range.

Parameter	Description
.AutoBlack	Lets you specify the current percentage of outlying pixels at the light end of the tonal range that Corel PHOTO-PAINT will ignore when performing an auto-equalization. The value ranges in steps from 1 to 1000; each step represents 0.05%.
.AutoWhite	Lets you specify the current percentage of outlying pixels at the dark end of the tonal range that Corel PHOTO-PAINT will ignore when performing an auto-equalization. The value ranges in steps from 1 to 1000; each step represents 0.05%.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageBCI (PHOTO-PAINT)

.ImageBCI .Brightness = *long*, .Contrast = *long*, .Intensity = *long*
.EndColorEffect

This command adjusts the brightness, contrast, and intensity of the tones in your image. The Brightness parameter shifts all pixel values up or down the tonal range. When you adjust the brightness, you lighten or darken all colors equally. The Contrast parameter adjusts the distance between your lightest and darkest pixels. When you increase the intensity, you brighten the lighter areas of your image without washing out the dark areas. Contrast and intensity usually go hand-in-hand. An increase in contrast sometimes washes out detail in shadows and highlights, and an increase in intensity can bring it back.

Parameter	Description
.Brightness	Lets you specify the Brightness of the current image. The value can range from -100 to 100.
.Contrast	Lets you specify the Contrast of the current image. The value can range from -100 to 100.
.Intensity	Lets you specify the Intensity of the current image. The value can range from -100 to 100.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageColorBalance (PHOTO-PAINT)

.ImageColorBalance .Red = *long*, .Green = *long*, .Blue = *long*, .Shadows = *boolean*, .Midtones = *boolean*, .Highlights = *boolean*, .Luminance = *boolean*
.EndColorEffect

This command shifts the colors in your image. This is useful for correcting color casts in your image. For example, if someone's face is too red in your photograph, you could shift values from red to cyan. You can also use the Color Balance filter to change the hue values for your entire image.

Parameter	Description
.Red	Lets you specify the balance of red in your image. The value can range from -100 to 100.
.Green	Lets you specify the balance of green in your image. The value can range from -100 to 100.
.Blue	Lets you specify the balance of blue in your image. The value can range from -100 to 100.
.Shadows	Set to TRUE (-1) to apply the color changes to the darkest pixels in the tonal range. Set to FALSE (0) to preserve the color of the dark pixels.
.Midtones	Set to TRUE (-1) to apply the color changes to the midtones of your image. Set to FALSE (0) to preserve the color of the midtone pixels.
.Highlights	Set to TRUE (-1) to apply the color changes to the lightest pixels in the tonal range. Set to FALSE (0) to preserve the color of the light pixels.
.Luminance	Set to TRUE (-1) to maintain the brightness values of your image. If you set this parameter to FALSE (0), the overall lightness or darkness of your image may be affected by color correction.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageColorCrop (PHOTO-PAINT)

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

This command crops 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.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.ToleranceMode	Lets you specify the tolerance mode: 0 = Normal 1 = HSB
.Normal	Lets you specify the tolerance as a percentage. This parameter is used only if .ToleranceMode is set to 0.
.Hue	Lets you specify the hue tolerance as a percentage. 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. This parameter is used only if .ToleranceMode is set to 1.
.Saturation	Lets you specify the saturation tolerance as a percentage. Saturation is the purity of a color. The more colors used to mix a color, the duller the color looks. This parameter is used only if .ToleranceMode is set to 1.
.Brightness	Lets you specify the brightness tolerance as a percentage. In the HSB color model, the component that determines the amount of black in a color. This parameter is used only if .ToleranceMode is set to 1.

Example

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

This example crops out a blue color border, using a normal tolerance of 0.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageColorTable (PHOTO-PAINT)

.ImageColorTable .Colors = *long*

This command changes the palette of a paletted image, using a series of PaletteColor commands inside a command block. ImageColorTable command blocks must end with an EndColorTable command.

Parameter	Description
.Colors	Lets you specify the number of colors in the palette. Valid values range from 1 to 256.

Example

```
.ImageColorTable 256
    .PaletteColor 5, 255, 0, 255, 0, 0
    .PaletteColor 5, 249, 0, 255, 0, 1
    ...
    .PaletteColor 5, 255, 255, 255, 0, 255
    .EndColorTable
```

This example changes the existing palette to a new palette containing 256 colors. A PaletteColor command is needed for each color in the new palette.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageColorTone (PHOTO-PAINT)

.ImageColorTone .Hue = *long*, .Saturation = *long*, .Lightness = *long*, .Brightness = *long*, .Contrast = *long*, .Intensity = *long*

This command changes the color tone of your image using one or both of the HSL and BCI color models.

Parameter	Description
.Hue	Lets you specify the hue of the current image.
.Saturation	Lets you specify the saturation of the current image.
.Lightness	Lets you specify the lightness of the current image.
.Brightness	Lets you specify the Brightness of the current image.
.Contrast	Lets you specify the Contrast of the current image.
.Intensity	Lets you specify the Intensity of the current image.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageConvert (PHOTO-PAINT)

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

This command converts the loaded image to another graphic format.

Parameter	Description
.Type	Lets you specify the image type: 1 = RGB 2 = 256 Grayscale 3 = Black and white 4 = Paletted 5 = CMYK 7 = LAB
.RenderType	Lets you specify the type of rendering to apply 1 = None 2 = Error Diffusion 3 = Ordered
.PaletteType	Lets you specify the palette type used (only applies when converting 256 Color images; parameter value ignored in other cases): 0 = Optimized 1 = Adaptive 2 = Uniform 4 = StdVGA
.Threshold	Darkens the image. Valid threshold values range from 1 to 255, and are only used to convert Black and white with a .RenderType equal to 1.
.HalftoneType	Lets you specify the Halftone types of the image (only applies when converting Black and white images; parameter value ignored in other cases): 0 = None 1 = Square 2 = Round 3 = Line 4 = Cross
.Angle	Lets you specify the angle of conversion, in degrees, to apply to the image (only applies when .HalftoneType does not equal 0; parameter value ignored in other cases). Positive numbers result in counter-clockwise rotation; negative numbers result in clockwise rotation.
.Width	Lets you specify the halftone width in pixels (only applies when .HalftoneType does not equal 0; parameter value ignored in other cases).
.Number	Lets you specify the number of colors in the palette for conversion to a paletted image. Values are 1 to 256.
.Flatten	0 does not merge objects to one object 1 merges objects to one object

Note

When converting with a custom palette, the user should use .ImageConvertPaletted. Use the ImageConvertDuotone command to convert to a duotone.

Example

```
.ImageConvert 4, 2, 100, 0, 0, 0, 0, 64
```

This example converts an image to a Paletted image using an optimized palette and error diffusion with 64 colors.

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

This example converts an image to grayscale.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageConvertDuotone (PHOTO-PAINT)

.ImageConvertDuotone .Style = *long*, .UseOverprints = *boolean*

This command converts a grayscale image to a duotone image. An ImageConvertDuotone command block contains several commands that set attributes of the conversion process, and must end with an EndConvertDuotone command.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.UseOverprints	Set to TRUE (-1) to use overprints.

Example

```
.ImageConvertDuotone 2, TRUE
    .OverprintColor 0, 2, 26, 97, 0
    .OverprintColor 1, 0, 100, 0, 100
    .OverprintColor 2, 100, 0, 0, 100
    ...
    .OverprintColor 10, 100, 100, 100, 100

    .DuotoneInfo 0, 3, 0, 0, 0, 255
    .DuotoneHandle 0, 0, 0, 0
    .DuotoneHandle 0, 1, 117, 66
    .DuotoneHandle 0, 2, 255, 100

    .DuotoneInfo 1, 4, 0, 0, 255, 0
    .DuotoneHandle 1, 0, 0, 0
    .DuotoneHandle 1, 1, 85, 73
    .DuotoneHandle 1, 2, 166, 40
    .DuotoneHandle 1, 3, 255, 100
    .EndConvertDuotone
```

This example converts a grayscale image to a duotone with black and yellow channels.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageConvertPaletted (PHOTO-PAINT)

.ImageConvertPaletted .RenderType = *long*, .Colors = *long*, .Flatten = *boolean*

This command converts the current document to a paletted image using the palette colors specified inside the command block. An ImageConvertPaletted command block must end with an EndConvertPaletted command.

Parameter	Description
.RenderType	Lets you specify the palette type: 1 = None 2 = Error Diffusion 3 = Ordered
.Colors	Lets you specify the number of colors to include in an Adaptive or Optimized palette. Additional colors will not be added if you select more colors than are used in the image. Black and white images are the exception: a palette with 256 shades of grays will be created on conversion. Valid values range from 1 to 256.
.Flatten	0 does not merge objects to one object 1 merges objects to one object

Example

```
.ImageConvertPaletted 1, 216
    .PaletteColor 5, 0, 0, 0, 0, 0
    .PaletteColor 5, 51, 0, 0, 0, 1
    .PaletteColor 5, 102, 0, 0, 0, 2
    .PaletteColor 5, 153, 0, 0, 0, 3
    ...
    .PaletteColor 5, 153, 255, 255, 0, 213
    .PaletteColor 5, 204, 255, 255, 0, 214
    .PaletteColor 5, 255, 255, 255, 0, 215
    .EndConvertPaletted
```

This example converts an image to a paletted image, mapping the existing colors to the new palette colors using a Uniform render.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageConvertVideoNTSC (PHOTO-PAINT)

.ImageConvertVideoNTSC

This command converts the current document to the NTSC RGB video color mode, using the colors in the NTSC RGB gamut. Convert your 24-bit RGB images to the video color mode to create images with colors that are suitable for television reproduction. This prevents oversaturation and retains the integrity of your image when it is broadcast.

Example

```
.ImageConvertVideoNTSC
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageCrop (PHOTO-PAINT)

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

This command crops the current image to the specified rectangle.

Parameter	Description
.Left	Lets you specify the X-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the image's bounding box in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the image's bounding box in pixels, relative to the origin.

Example

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

This example crops the image to the specified rectangle.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageCropToMask (PHOTO-PAINT)

.ImageCropToMask

This command crops the selected image to the bounding box of the current mask.

Note

A mask must be present before this command can be used.

Example

```
.ImageCropToMask
```

This example crops the selected image to the size of the current mask.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageDeInterlace (PHOTO-PAINT)

.ImageDeInterlace .ReplaceMode = *long*

This command removes even or odd horizontal lines from scanned or interlaced video images. You can fill the spaces left by the discarded lines using either of two methods: duplication fills in the spaces with copies of the adjacent lines of pixels, while interpolation fills them in with colors created by averaging the surrounding pixels.

Parameter	Description
.ReplaceMode	Lets you specify the deinterlace fill method: 0 = Replaces even lines using the duplication method 1 = Replaces even lines using the interpolation method 2 = Replaces odd lines using the duplication method 3 = Replaces odd lines using the interpolation method

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageDesaturate (PHOTO-PAINT)

.ImageDesaturate
.EndColorEffect

This command reduces the saturation of each color in your image to 0, which converts each color to its grayscale equivalent. This makes your image appear to be grayscale without having to convert it.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageDeskew (PHOTO-PAINT)

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

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

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Note

The color parameters specify the paper or border color around the image, which is used to the find the image edges

Example

```
.ImageDeskew 3, 0, 119, 211,0
```

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

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageDeskewCrop (PHOTO-PAINT)

.ImageDeskewCrop .Angle = *double*, .Width = *long*, .Height = *long*, .PointX = *long*, .PointY = *long*

This command crops an image. If the angle is set to a value other than 0, the image is also deskewed.

Parameter	Description
.Angle	Lets you specify the angle of rotation. Values range from -360 - 360 degrees.
.Width	Lets you specify the width of the crop marquee.
.Height	Lets you specify the height of the crop marquee.
.PointX	Lets you specify the horizontal coordinate for the left edge of the cropped area.
.PointY	Lets you specify the vertical coordinate for the top of the cropped area.

Example

```
.ImageDeskewCrop -11.7, 200, 200, 264, 140
```

This example crops a rectangle to 200 pixels by 200 pixels and skews it 11.7 degrees. The center point is at 264,140.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageDuplicate (PHOTO-PAINT)

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

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

Parameter	Description
.FileName	Lets you specify the new name of the duplicated image.
.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
```

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

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageEqualize (PHOTO-PAINT)

.ImageEqualize .Method = *long*, .AutoBlack = *long*, .AutoWhite = *Long*, .AutoAdjust = *boolean*

This command equalizes the current image. An ImageEqualize command block must end with an EndImageEqualize command.

Parameter	Description
.Method	Lets you specify the equalization method: 0 = Proportional 1 = Nonproportional
.AutoBlack	Lets you specify the percentage of outlying pixels at the dark end of the tonal range that Corel PHOTO-PAINT will ignore when performing an auto-equalization. Valid values range from 0 to 100%.
.AutoWhite	Lets you specify the percentage of outlying pixels at the light end of the tonal range that Corel PHOTO-PAINT will ignore when performing an auto-equalization. Valid values range from 0 to 100%.
.AutoAdjust	Set to TRUE (-1) to let Corel PHOTO-PAINT determine the .AutoBlack and .AutoWhite parameters.

Example

```
.ImageEqualize 0, 5, 5, FALSE
    .ImageEqualizeChannel 0, 27, 255, 8, 249, 208
    .ImageEqualizeChannel 1, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 2, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 3, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 4, 0, 255, 0, 255, 100
    .EndImageEqualize
```

This example equalizes the overall RGB channel without enhancing any of the channels individually.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageEqualizeChannel (PHOTO-PAINT)

.ImageEqualizeChannel .Index = *long*, .InLow = *long*, .InHigh = *long*, .OutLow = *long*, .OutHigh = *long*, .Gamma = *long*

This command sets the channel equalization attributes for the ImageEqualize command. This command must appear inside an ImageEqualize command block.

Parameter	Description
.Index	Lets you specify the equalization channel: 0 = RGB channels 1 = Red channel 2 = Green channel 3 = Blue channel
.InLow	Lets you specify a clipping range for the darkest pixels in your image. All pixels that fall between this value and the .OutLow value will map to the darkest pixel value. Valid values range from 0 to 254.
.InHigh	Lets you specify a clipping range for the brightest pixels in your image. All pixels that fall between this value and the .OutHigh value will map to the brightest pixel value. Valid values range from 1 to 255.
.OutLow	Lets you specify the output brightness value of the darkest pixels in your image. Valid values range from 0 to 254.
.OutHigh	Lets you specify the output brightness value of the lightest pixels in your image. Valid values range from 1 to 255.
.Gamma	Lets you specify the gamma curve value. Adjusting the gamma curve value allows you to pick up detail in a low contrast image without significantly affecting the shadows or highlights. It does affect all the values in your image, but is curve-based so the changes are weighted toward the midtones. Valid values range from 1 to 1000.

Example

```
.ImageEqualize 0, 5, 5, FALSE
    .ImageEqualizeChannel 0, 27, 255, 8, 249, 208
    .ImageEqualizeChannel 1, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 2, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 3, 0, 255, 0, 255, 100
    .ImageEqualizeChannel 4, 0, 255, 0, 255, 100
    .EndImageEqualize
```

This example equalizes the overall RGB channel without enhancing any of the channels individually.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageFlipHorizontal (PHOTO-PAINT)

.ImageFlipHorizontal

This command flips the image horizontally.

Example

```
.ImageFlipHorizontal
```

This example flips the active image horizontally.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageFlipVertical (PHOTO-PAINT)

.ImageFlipVertical

This command flips the image vertically.

Example

```
.ImageFlipVertical
```

This example flips the active image vertically.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageGamma (PHOTO-PAINT)

.ImageGamma .Value = *long*
.EndColorEffect

This command picks out detail in low contrast images without significantly affecting shadows or highlights. It does affect all the values in your image, but is curve-based so that the changes are weighted toward the midtones.

Parameter	Description
.Value	Lets you specify the degree of gamma correction. Valid values range from 0.1 to 10.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageHSL (PHOTO-PAINT)

.ImageHSL .Hue = *long*, .Saturation = *long*, .Lightness = *long*
.EndColorEffect

This command adjust the colors in your image using HLS (Hue, Lightness, and Saturation) values

Parameter	Description
.Hue	Lets you specify the Hue tolerance. 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. Valid values range from -100 to 100.
.Saturation	Lets you specify the Saturation tolerance. 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. Valid values range from -100 to 100.
.Lightness	Lets you specify the Lightness tolerance. Lightness is the amount of black or white in a color. Valid values range from -100 to 100.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageHSLChannel (PHOTO-PAINT)

.ImageHSLChannel .Channel = *long*, .Hue = *long*, .Saturation = *long*, .Lightness = *long*

This command specifies the HSL values for each channel in an image. The .ImageHSL command sets the values for the master channel.

Parameter	Description
.Channel	Lets you specify the channel 1 = red 2 = yellow 3 = green 4 = cyan 5 = blue 6 = magenta 7 = Grayscale
.Hue	Lets you specify the Hue tolerance. 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. Valid values range from -180 to 180.
.Saturation	Lets you specify the Saturation tolerance. 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. Valid values range from -100 to 100.
.Lightness	Lets you specify the Lightness tolerance. Lightness is the amount of black or white in a color. Valid values range from -100 to 100.

Example

```
.ImageHSL -46, 48, -29
        .ImageHSLChannel 1, 180, -100, -100
        .ImageHSLChannel 2, 0, -18, 25
        .ImageHSLChannel 3, 0, 34, 47
        .ImageHSLChannel 4, 69, -32, 0
        .ImageHSLChannel 5, 0, 19, 40
        .ImageHSLChannel 6, -125, 0, 24
        .ImageHSLChannel 7, -110, 22, -31
        .EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageInvert (PHOTO-PAINT)

.ImageInvert
.EndColorEffect

This command inverts the colors in your image, producing an effect much like a photographic negative.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageLevelThreshold (PHOTO-PAINT)

.ImageLevelThreshold .Channel = *long*, .Low = *long*, .Threshold = *long*, .High = *Long*, .BiLevel = *long*
.EndColorEffect

This command converts certain shades of each color in an image to black or white. In bi-level mode, the command can convert shades to both black and white at the same time.

Parameter	Description
.Channel	Lets you specify the channel to convert: 0 = All three channels (RGB) at once 1 = Red channel 2 = Green channel 3 = Blue Channel
.Low	Lets you specify the low-level value. Valid values range from 0 to 255.
.Threshold	Lets you specify the threshold value. Valid values range from 0 to 255.
.High	Lets you specify the high-level value. Valid values range from 0 to 255.
.BiLevel	Lets you specify the conversion method: 0 = To black 1 = To White 2 = Bi-level

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImagePapersize (PHOTO-PAINT)

.ImagePapersize .Width = *long*, .Height = *long*, .Xoffset = *long*, .Yoffset = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command adjusts 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.

Parameter	Description
.Width	Lets you specify the paper width in pixels.
.Height	Lets you specify the paper height in pixels.
.Xoffset	Lets you specify the placement of the image on the paper offset from the x-axis (expressed in pixels).
.Yoffset	Lets you specify the placement of the image on the paper offset from the y-axis (expressed in pixels).
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.ImagePapersize 408, 528, 0, 0, 5, 255, 200, 100, 0
```

This example sets the paper size to 408 pixels (width) by 528 pixels (height) with no offset applied.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImagePosterize (PHOTO-PAINT)

.ImagePosterize .Level = *long*
.EndColorEffect

This command transforms the color range of your image to solid blocks of color, reducing gradual blends to hard edges between areas of color.

Parameter	Description
.Level	Lets you specify the intensity of the posterization effect (i.e. the number of colors in the final image). Valid values range from 2 to 32.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageReplaceColors (PHOTO-PAINT)

.ImageReplaceColors .OrgH = *long*, .OrgS = *long*, .OrgL = *long*, .DestH = *long*, .DestS = *long*, .DestL = *long*, .Range = *long*, .IgnoreGrayscale = *bool*
.EndColorEffect

This command replaces one color in your image with another color. Depending on the .Range value, the command will replace a single color, or shift the entire image from one range of color to another.

Parameter	Description
.OrgH	The Hue value of the original color. Valid values range from 0 to 6144.
.OrgS	The Saturation value of the original color. Valid values range from 0 to 1024.
.OrgL	The Lightness value of the original color. Valid values range from 0 to 1024.
.DestH	The Hue value of the destination color. Valid values range from -180 to 180.
.DestS	The Saturation value of the destination color. Valid values range from -100 to 100.
.DestL	The Lightness value of the destination color. Valid values range from -100 to 100.
.Range	The range of color that will be replaced by the command. Valid values range from 1 to 100.
.IgnoreGrayscale	Determines whether grayscale values are ignored.

Example

```
.ImageReplaceColors 3460, 562, 795, -150, 100, 75, 34, 1
.EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageResample (PHOTO-PAINT)

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

This command adjusts the dimensions and resolution of an image.

Parameter	Description
.Width	Lets you specify the new width of the image in pixels.
.Height	Lets you specify the new height of the image in pixels.
.HRes	Lets you specify the new horizontal resolution of the image in dots per inch (dpi).
.VRes	Lets you specify 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
```

This 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(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageRotate (PHOTO-PAINT)

.ImageRotate .Angle = *double*, .Clip = *boolean*, .AntiAlias = *boolean*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command rotates the image a specified angle.

Parameter	Description
.Angle	Lets you specify the 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.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.ImageRotate 315, TRUE, TRUE, 5, 255, 204, 53, 0
```

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

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageSelectiveColor (PHOTO-PAINT)

.ImageSelectiveColor .Mode = *long*

This command applies the Selective Color effects.

Parameter	Description
.Mode	Lets you specify the mode 1 = Relative 0 = Absolute

Example

```
.ImageSelectiveColor 1
        .ImageSelectiveColorChannel 0, 0, 30, 42, -20
        .ImageSelectiveColorChannel 1, 19, 15, 17, 3
        .ImageSelectiveColorChannel 2, 30, 0, 28, 0
        .ImageSelectiveColorChannel 3, -42, 0, 29, 29
        .ImageSelectiveColorChannel 4, 50, 33, 0, -44
        .ImageSelectiveColorChannel 5, -47, -46, 0, 30
        .ImageSelectiveColorChannel 6, 0, 0, 0, 10
        .ImageSelectiveColorChannel 7, 0, 0, 0, -29
        .ImageSelectiveColorChannel 8, 0, 0, 0, 28
        .EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageSelectiveColorChannel (PHOTO-PAINT)

.ImageSelectiveColorChannel .Channel = *long*, .Cyan = *long*, .Magenta = *long*, .Yellow = *long*, .Black = *long*

This command specifies the color channel used to apply the Selective Color effect.

Parameter	Description
.Channel	Lets you specify the channel 0 = red 1 = yellow 2 = green 3 = cyan 4 = blue 5 = magenta 6 = shadow 7 = midtone 8 = highlight
.Cyan	Lets you specify the cyan color value. Values range from -100 - 100.
.Magenta	Lets you specify the magenta color value. Values range from -100 - 100.
.Yellow	Lets you specify the yellow color value. Values range from -100 - 100.
.Black	Lets you specify the black color value. Values range from -100 - 100.

Example

```
.ImageSelectiveColor 1
.ImageSelectiveColorChannel
.EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageSetChannel (PHOTO-PAINT)

.ImageSetChannel .Channel = *long*

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.

Parameter	Description
.Channel	Lets you specify 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
```

This example selects the red image channel to be edited.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageSplit (PHOTO-PAINT)

.ImageSplit .Type = *long*

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

Parameter	Description
.Type	Lets you specify how to split the image into channels: 0 = split the image into RGB channels 1 = split the image into CMYK channels 2 = split the image into HSB channels 3 = split the image into HLS channels 4 = split the image into YIQ channels 5 = split the image into LAB channels

Note

- RGB images can be split into all color models except LAB. LAB images can only be split into LAB channels, and CMYK images can only be split to CMYK channels.

Example

```
.ImageSplit 4
```

This example splits an RGB image into YIQ channels.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageSprayerSettings (PHOTO-PAINT)

.ImageSprayerSettings .Dabs = *long*, .Spacing = *long*, .Spread = *long*, .FadeOut = *long*, .Type = *long*, .From = *long*, .To = *long*, .Start = *long*

This command sets the brush stroke attributes of the Image Sprayer tool. This command must appear in an ImageSprayerTool command block.

Parameter	Description
.Dabs	Lets you specify the number of dabs in the brush stroke. Valid values range from 1 to 25.
.Spacing	Lets you specify the spacing between dabs along the length of the stroke. Valid values range from 1 to 999 pixels.
.Spread	Lets you specify the spacing between dabs along the width of the stroke. Valid values range from 1 to 999 pixels.
.FadeOut	Lets you specify the length of the brush stroke before it fades out. Valid values range from 0 to 100.
.Type	Lets you specify the order in which the images will be sprayed in each brush stroke: 0 = Random 1 = Sequential 2 = Directional
.From	The number of the first image in the sequence.
.To	The number of the last image in the sequence.
.Start	Lets you specify the image in the sequence with which to start painting.

Example

```
.ImageSprayerTool "fruit.cpt", 1, 5, 87, 0, 0
    .ImageSprayerSettings 1, 11, 19, 0, 0, 1, 5, 1
    .RandomSeed 552585248
    .StartDraw 11904, 8320, 0, 0
    .ContinueDraw 14343, 9096, 0, 0
    ...
    .ContinueDraw 61761, 18445, 0, 0
    .EndDraw
```

This example draws a brush stroke containing a variety of fruits.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageSprayerTool (PHOTO-PAINT)

.ImageSprayerTool .FileName = *string*, .Row = *long*, .Column = *long*, .Size = *long*, .MergeMode = *long*, .Transparency = *long*

This command paints a variety of images from an image file along a brush stroke. An ImageSprayerTool command block must contain a series of StartDraw and ContinueDraw commands, and end with an EndDraw command.

Parameter	Description
.FileName	Lets you specify the name of the image list to use.
.Row	Lets you specify the number of rows of images.
.Column	Lets you specify the number of columns of images.
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.MergeMode	Lets you specify 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
.Transparency	Lets you specify the transparency level of the nib. Valid values range from 0 to 99%.

Example

```
.ImageSprayerTool "fruit.cpt", 1, 5, 87, 0, 0
    .ImageSprayerSettings 1, 11, 19, 0, 0, 1, 5, 1
    .RandomSeed 552585248
    .StartDraw 11904, 8320, 0, 0
    .ContinueDraw 14343, 9096, 0, 0
    ...
    .ContinueDraw 61761, 18445, 0, 0
    .EndDraw
```

This example draws a brush stroke containing a variety of fruits.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageSTBalance (PHOTO-PAINT)

.ImageSTBalance .Channel = *long*, .UseLow = *boolean*, .UseMid = *boolean*, .UseHigh = *boolean*, .UseAll = *boolean*

This command changes a specific color in your image to a target color. An ImageSTBalance command block must contain a series of ImageSTColor commands, and end with an EndImageSTBalance.

Parameter	Description
.Channel	Lets you specify the channel you want to change: 0 = All three channels (RGB) at once 1 = Red channel 2 = Green channel 3 = Blue Channel
.UseLow	Set to TRUE (-1) to use the low point color values set with the ImageSTColor commands in the command block.
.UseMid	Set to TRUE (-1) to use the mid point color values set with the ImageSTColor commands in the command block.
.UseHigh	Set to TRUE (-1) to use the high point color values set with the ImageSTColor commands in the command block.
.UseAll	Set to TRUE (-1) to use all of the color values set with the ImageSTColor commands in the command block.

Example

```
.ImageSTBalance 0, TRUE, TRUE, TRUE, TRUE
    ' Source low color
    .ImageSTColor 0, 5, 249, 249, 153, 0
    ' target low color
    .ImageSTColor 1, 5, 249, 249, 153, 0
    ' source mid color
    .ImageSTColor 2, 5, 249, 249, 222, 0
    ' target mid color
    .ImageSTColor 3, 5, 249, 249, 153, 0
    ' source high color
    .ImageSTColor 4, 5, 222, 189, 222, 0
    ' target high color
    .ImageSTColor 5, 5, 222, 189, 222, 0
    .EndImageSTBalance
```

This example maps each of the low, mid, and high colors to the designated target colors.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageSTColor (PHOTO-PAINT)

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

This command sets the source and target colors for the ImageSTBalance command. This command must occur in an ImageSTBalance command block.

Parameter	Description
.Index	Lets you specify the source or target colors to define: 0 = Source low color 1 = Target low color 2 = Source mid color 3 = Target mid color 4 = Source high color 5 = Target high color
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Note

- The order of the ImageSTColor commands is important. The definitions of source and target commands must occur in pairs for the command to work.

Example

```
.ImageSTBalance 0, TRUE, TRUE, TRUE, TRUE
    ' Source low color
    .ImageSTColor 0, 5, 249, 249, 153, 0
    ' target low color
    .ImageSTColor 1, 5, 249, 249, 153, 0
    ' source mid color
    .ImageSTColor 2, 5, 249, 249, 222, 0
    ' target mid color
    .ImageSTColor 3, 5, 249, 249, 153, 0
    ' source high color
    .ImageSTColor 4, 5, 222, 189, 222, 0
    ' target high color
    .ImageSTColor 5, 5, 222, 189, 222, 0
    .EndImageSTBalance
```

This example maps each of the low, mid, and high colors to the designated target colors.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageToneCurve (PHOTO-PAINT)

.ImageToneCurve

This command adjusts the tone curve of the current image.

If you want to apply the adjustment directly to the image pixels, add the ImageToneTable command to it, and end the command block with the EndImageToneCurve command.

On the other hand, if you want to apply the adjustment to a lens object, you must add the ImageToneChannel and ImageTonePoint commands, and end the command block with the EndColorEffect command. Examples are provided for both options.

Example when applying to the image

```
.ImageToneCurve
    .ImageToneTable 0, 0, 0, 0, 0
    .ImageToneTable 1, 1, 0, 1, 0
    .ImageToneTable 2, 1, 0, 1, 1
    .ImageToneTable 3, 2, 0, 2, 1
    ...
    .ImageToneTable 254, 255, 252, 255, 253
    .ImageToneTable 255, 255, 255, 255, 255
    .EndImageToneCurve
```

This example defines a new tone curve for the current image. Each of the 256 ImageToneTable commands defines one tone in the new tone curve.

Example when applying to a lens

```
.ImageToneCurve
    .ImageToneChannel 0, 2, 100, 256, 0, 1
    .ImageTonePoint 0, 0, 0, 0
    .ImageTonePoint 0, 1, 255, 255
    .ImageToneChannel 1, 2, 100, 256, 0, 1
    .ImageTonePoint 1, 0, 0, 0
    .ImageTonePoint 1, 1, 255, 255
    .ImageToneChannel 2, 2, 100, 256, 0, 1
    .ImageTonePoint 2, 0, 0, 0
    .ImageTonePoint 2, 1, 255, 255
    .ImageToneChannel 3, 2, 100, 256, 0, 1
    .ImageTonePoint 3, 0, 0, 0
    .ImageTonePoint 3, 1, 255, 255
    .ImageToneChannel 4, 4, 100, 256, 0, 1
    .ImageTonePoint 4, 0, 0, 0
    .ImageTonePoint 4, 1, 142, 72
    .ImageTonePoint 4, 2, 166, 173
    .ImageTonePoint 4, 3, 255, 255
    .LensNew "Tone Curve Lens 5"
    .EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageToneTable (PHOTO-PAINT)

.ImageToneTable .Number = *long*, .Curve1 = *long*, .Curve2 = *long*, .Curve3 = *long*, .Curve4 = *long*

This command sets the color of a single tone on the tone curve of the current image. This command must appear in an ImageToneCurve command block.

Parameter	Description
.Number	Lets you specify the index number of the individual tone to change.
.Curve1	Lets you specify the first color component of the tone using the current color model. For example, Red is the first color component for RGB.
.Curve2	Lets you specify the first color component of the tone using the current color model. For example, Green is the second color component for RGB. If this parameter is not available in the current color model, set it to 0.
.Curve3	Lets you specify the first color component of the tone using the current color model. For example, Blue is the third color component for RGB. If this parameter is not available in the current color model, set it to 0.
.Curve4	Lets you specify the first color component of the tone using the current color model. For example, Black is the fourth color component for CMYK. If this parameter is not available in the current color model, set it to 0.

Example

```
.ImageToneCurve
    .ImageToneTable 0, 0, 0, 0, 0
    .ImageToneTable 1, 1, 0, 1, 0
    .ImageToneTable 2, 1, 0, 1, 1
    .ImageToneTable 3, 2, 0, 2, 1
    ...
    .ImageToneTable 254, 255, 252, 255, 253
    .ImageToneTable 255, 255, 255, 255, 255
    .EndImageToneCurve
```

This example defines a new tone curve for the current image. Each of the 256 ImageToneTable commands defines one tone in the new tone curve.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ImageToneChannel (PHOTO-PAINT)

.ImageToneChannel .Channel=*long*, .Nodes=*long*, .Gamma=*long*, Length=*long*, .DrawMode=*long*, .Orientation=*long*

This command sets the overall attributes of the tone curve for a specific channel when applying the Tone Curve to a lens object. This command must appear in the ImageToneCurve command block when a new Tone Curve is used on a lens.

Parameter	Description
.Channel	Lets you specify the channel to which the next parameters apply 0 = Red or Cyan channel 1 = Green or Magenta channel 2 = Blue or Yellow channel 3 = Black 4 = All channels
.Nodes	Lets you specify the number of nodes on the tone curve for the selected channel Maximum number: 256
.Gamma	Lets you specify the gamma value for the channel. Gamma values range from 0 to 1000
.Length	Must always be 256
.DrawMode	Lets you specify the Edit Style selected for the tone curve 0 = Curve style 1 = Linear style 2 = Freehand style 3 = Gamma style
.Orientation	Determines the orientation of the horizontal and vertical axes of the tone curve graph 0 = the brightness value of the shadows are near the origin point 1 = the brightness value of the highlights are near the origin point

Example

```
.ImageToneCurve
    .ImageToneChannel 0, 2, 100, 256, 0, 1
    .ImageTonePoint 0, 0, 0, 0
    .ImageTonePoint 0, 1, 255, 255
    .ImageToneChannel 1, 2, 100, 256, 0, 1
    .ImageTonePoint 1, 0, 0, 0
    .ImageTonePoint 1, 1, 255, 255
    .ImageToneChannel 2, 2, 100, 256, 0, 1
    .ImageTonePoint 2, 0, 0, 0
    .ImageTonePoint 2, 1, 255, 255
    .ImageToneChannel 3, 2, 100, 256, 0, 1
    .ImageTonePoint 3, 0, 0, 0
    .ImageTonePoint 3, 1, 255, 255
    .ImageToneChannel 4, 4, 100, 256, 0, 1
    .ImageTonePoint 4, 0, 0, 0
    .ImageTonePoint 4, 1, 142, 72
    .ImageTonePoint 4, 2, 166, 173
    .ImageTonePoint 4, 3, 255, 255
    .LensNew "Tone Curve Lens 5"
    .EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ImageTonePoint (PHOTO-PAINT)

.ImageTonePoint .Channel=*long*, .Index=*long*, .X=*long*, .Y=*long*

This command identifies the precise location of each node found on a tone curve that will be applied to a lens object. This command must appear in the ImageToneCurve command block when a new Tone Curve is used on a lens.

Parameter	Description
.Channel	Lets you specify the channel for which you are setting the node location 0 = Red or Cyan channel 1 = Green or Magenta channel 2 = Blue or Yellow channel 3 = Black 4 = All channels
.Index	Lets you specify the number of the node (the maximum number is 256)
.X	Sets the location of the node on the horizontal axis (from 0 to 256)
.Y	Sets the location of the node on the vertical axis (from 0 to 256)

Example

```
.ImageToneCurve
    .ImageToneChannel 0, 2, 100, 256, 0, 1
    .ImageTonePoint 0, 0, 0, 0
    .ImageTonePoint 0, 1, 255, 255
    .ImageToneChannel 1, 2, 100, 256, 0, 1
    .ImageTonePoint 1, 0, 0, 0
    .ImageTonePoint 1, 1, 255, 255
    .ImageToneChannel 2, 2, 100, 256, 0, 1
    .ImageTonePoint 2, 0, 0, 0
    .ImageTonePoint 2, 1, 255, 255
    .ImageToneChannel 3, 2, 100, 256, 0, 1
    .ImageTonePoint 3, 0, 0, 0
    .ImageTonePoint 3, 1, 255, 255
    .ImageToneChannel 4, 4, 100, 256, 0, 1
    .ImageTonePoint 4, 0, 0, 0
    .ImageTonePoint 4, 1, 142, 72
    .ImageTonePoint 4, 2, 166, 173
    .ImageTonePoint 4, 3, 255, 255
    .LensNew "Tone Curve Lens 5"
    .EndColorEffect
```

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PaletteColor (PHOTO-PAINT)

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

This command sets a color in a custom palette. This command must appear as part of a block, and is used by such commands as ImageColorTable and ImageConvertPaletted.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.Index	Lets you specify the position of the color in the palette. Since .Index is 0-based, the first color is 0, the second color is 1, and so on.

Example

```
.ImageColorTable 256
    .PaletteColor 5, 255, 0, 255, 0, 0
    .PaletteColor 5, 249, 0, 255, 0, 1
    ...
    .PaletteColor 5, 255, 255, 255, 0, 255
    .EndColorTable
```

This example changes the existing palette to a new palette containing 256 colors. A PaletteColor command is needed for each color in the new palette.

{button ,AL(`OVR1 Image commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Color Mask commands (PHOTO-PAINT)

ColorMaskColor (PHOTO-PAINT)

.ColorMaskColor .Number = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Normal = *long*, .Hue = *long*, .Saturation = *long*, .Brightness = *long*

This command sets color attributes for the ColorMaskCreateMask and ColorMaskCreateChannel commands. This command must appear in a ColorMaskCreateMask or ColorMaskCreateChannel command block.

Parameter	Description
.Number	Identifies the colors in the Color Mask. Valid values range from 1 to 10.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.Normal	Lets you specify the Normal-mode tolerance. Valid values range from 0 to 100%.
.Hue	Lets you specify the hue tolerance. 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. Valid values range from 0 to 100%.
.Saturation	Lets you specify the saturation tolerance. 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. Valid values range from 0 to 100%.
.Brightness	Lets you specify the brightness tolerance. In the HSB color model, the component that determines the amount of black in a color. Valid values range from 0 to 100%.

Example

```
.ObjectColorTransparencyTool 0, TRUE, 27
    .ColorMaskColor 0, 5, 62, 155, 0, 0, 14, 30, 50, 10
    .ColorMaskColor 1, 5, 62, 155, 0, 0, 14, 30, 50, 10
    .ColorMaskColor 2, 5, 112, 102, 28, 0, 14, 30, 50, 10
    .EndObjectColorTransparencyTool
```

This example makes the selected colors transparent in the currently selected object using a clip mask.

{button ,AL(`OVR1 Color Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ColorMaskCreateChannel (PHOTO-PAINT)

.ColorMaskCreateChannel .MaskMode = *long*, .Smoothing = *long*, .ToleranceMode = *long*, .Gamut = *boolean,* .Threshold = *long,* .BrightnessModel = *long,* .Name = *string*

This command creates a new color channel using colors defined with a series of ColorMaskColor commands. A ColorMaskCreateMask command block must end with an EndColorMask command.

Parameter	Description
.MaskMode	Lets you specify the mode of the channel 0 = Protect 1 = Modify
.Smoothing	Lets you specify the amount of smoothing to be used in the channel. Valid values range from 0 to 100.
.ToleranceMode	Lets you specify the tolerance mode to set: 0 = Normal 1 = HSB 2 = Hue 3 = Saturation 4 = Brightness
.Gamut	Set to TRUE (-1) to enable gamut control.
.Threshold	Converts selected colors to black. This command uses black to display the pixels with a brightness value that is below the threshold that you set. Masked areas are always displayed in black however, black may translate differently depending on the preview method you specify. Values range from 0 to 255.
.BrightnessModel	Converts selected colors to white. This command uses white to display the pixels with a brightness value that is above the threshold you set. Selected areas on an image are always displayed in white however, this may translate differently depending on the preview method you specify. 0=Black 1=White
.Name	Lets you specify the name of the color channel that is created.

Example

```
.ColorMaskCreateChannel 1, 100, 2, FALSE, 0, 0, "Alpha 1"
    .ColorMaskColor 0, 5, 0, 0, 0, 0, 18, 18, 18, 18
    .EndColorMask
```

{button ,AL(`OVR1 Color Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ColorMaskCreateMask (PHOTO-PAINT)

.ColorMaskCreateMask .DrawMode = *long*, .MaskMode = *long*, .Smoothing = *long*, .ToleranceMode = *long*, .Gamut = *boolean,* .Threshold = *long,* .BrightnessModel = *long*

This command creates a mask from a color mask, using colors defined with a series of ColorMaskColor commands. A ColorMaskCreateMask command block must end with an EndColorMask command.

Parameter	Description
.DrawMode	Lets you specify the draw mode: 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.MaskMode	Lets you specify the mode of the Color Mask: 0 = Protect 1 = Modify
.Smoothing	Lets you specify the amount of smoothing to be applied by the Color Mask. Valid values range from 0 to 100.
.ToleranceMode	Lets you specify the tolerance mode to set: 0 = Normal 1 = HSB 2 = Hue 3 = Saturation 4 = Brightness
.Gamut	Set to TRUE (-1) to enable gamut control.
.Threshold	Converts selected colors to black. This command uses black to display the pixels with a brightness value that is below the threshold that you set. Masked areas are always displayed in black ▶ however, black may translate differently depending on the preview method you specify. Values range from 0 to 255.
.BrightnessModel	Converts selected colors to white. This command uses white to display the pixels with a brightness value that is above the threshold you set. Selected areas on an image are always displayed in white ▶ however, this may translate differently depending on the preview method you specify. 0=Black 1=White

Example

```
.ColorMaskCreateMask 0, 1, 100, 2, FALSE, 0, 0
    .ColorMaskColor 0, 5, 0, 0, 0, 0, 18, 18, 18, 18
    .ColorMaskColor 1, 5, 0, 0, 0, 0, 20, 20, 20, 20
    .ColorMaskColor 2, 5, 0, 0, 0, 0, 20, 20, 20, 20
    .EndColorMask
```

This example creates a color mask.

{button ,AL(`OVR1 Color Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ColorMaskReset (PHOTO-PAINT)

.ColorMaskReset

This command resets the color and tolerance settings to be used in the next call to the ColorMaskCreateChannel or ColorMaskCreateMask commands.

Example

```
.ColorMaskReset
```

This example resets the color mask settings for the next call to the ColorMaskCreateChannel or ColorMaskCreateMask commands.

{button ,AL(`OVR1 Color Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndColorMask (PHOTO-PAINT)

.EndColorMask

This command ends a ColorMaskCreateMask command block.

Example

```
.ColorMaskCreateMask 0, 0, 0, 0, FALSE
    .ColorMaskColor 0, 5, 245, 149, 180, 0, 10, 10, 10, 10
    .ColorMaskColor 1, 5, 245, 232, 180, 0, 10, 10, 10, 10
    .ColorMaskColor 2, 5, 245, 149, 180, 0, 10, 10, 10, 10
    .EndColorMask
```

This example creates a color mask.

{button ,AL(`OVR1 Color Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

OverprintColor (PHOTO-PAINT)

.OverprintColor .Number = *long*, .Cyan = *long*, .Magenta = *long*, .Yellow = *long*, .Black = *long*

This command sets the overprint colors for the ImageConvertDuotone command. This command must appear in an ImageConvertDuotone command block.

Parameter	Description
.Number	Lets you specify the index number of the overprint color.
.Cyan	Lets you specify the cyan color channel value of the overprint color. Valid values range from 0 to 100.
.Magenta	Lets you specify the magenta color channel value of the overprint color. Valid values range from 0 to 100.
.Yellow	Lets you specify the yellow color channel value of the overprint color. Valid values range from 0 to 100.
.Black	Lets you specify the black color channel value of the overprint color. Valid values range from 0 to 100.

Example

```
.ImageConvertDuotone 2, TRUE
    .OverprintColor 0, 2, 26, 97, 0
    .OverprintColor 1, 0, 100, 0, 100
    .OverprintColor 2, 100, 0, 0, 100
    ...
    .OverprintColor 10, 100, 100, 100, 100

    .DuotoneInfo 0, 3, 0, 0, 0, 255
    .DuotoneHandle 0, 0, 0, 0
    .DuotoneHandle 0, 1, 117, 66
    .DuotoneHandle 0, 2, 255, 100

    .DuotoneInfo 1, 4, 0, 0, 255, 0
    .DuotoneHandle 1, 0, 0, 0
    .DuotoneHandle 1, 1, 85, 73
    .DuotoneHandle 1, 2, 166, 40
    .DuotoneHandle 1, 3, 255, 100
    .EndConvertDuotone
```

This example converts a grayscale image to a duotone with black and yellow channels, using the ten defined overprint colors.

{button ,AL(`OVR1 Color Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Mask channel commands (PHOTO-PAINT)

AlphaOrder (PHOTO-PAINT)

.AlphaOrder .CurrentIndex = *long*, .NewIndex = *long*

This command reorders the mask or alpha channels in the Channels Docker window.

Parameter	Description
.CurrentIndex	Lets you specify the position of the current mask or alpha channel. 0 is the position of the first mask or alpha channel.
.NewIndex	Lets you specify the new position of the mask or alpha channel.

Example

```
.AlphaOrder 2, 0
```

This example reorders the alpha channels in the Channels Docker window.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ChannelNew (PHOTO-PAINT)

.ChannelNew .Name = *string*, .Opacity = *long*, .Invert = *long*, .FillWhite = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command creates a new alpha channel.

Parameter	Description
.Name	Lets you specify the name of the channel.
.Opacity	Lets you specify the opacity of the mask overlay that is applied to the channel.
.Invert	Lets you specify whether to invert the mask overlay that is applied to the channel.
.FillWhite	Set to TRUE (-1) to fill the mask selection with white. Set to FALSE (0) to fill the mask selection with black.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.ChannelNew "Alpha1", 50, 1, 0, 5, 255, 0, 0, 0
```

This example creates an alpha channel named "Alpha1". The overlay is red, inverted, and has 50% opacity.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ChannelProperties (PHOTO-PAINT)

.ChannelProperties .ChannelID = *long*, .Name = *string*, .Opacity = *long*, .Invert = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command sets the properties of an alpha channel

Parameter	Description
.ChannelID	Lets you specify the index number of the channel
.Name	Lets you specify the name of the channel.
.Opacity	Lets you specify the opacity of the mask overlay that is applied to the channel.
.Invert	Lets you specify whether to invert the mask overlay that is applied to the channel.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.ChannelProperties 2, "Sample", 25, 0, 5, 153, 255, 0, 0
```

This example changes the properties of an alpha channel named "Sample".

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskChannelAdd (PHOTO-PAINT)

.MaskChannelAdd .MaskName = *string*

This command saves a mask as a mask channel.

Parameter	Description
.MaskName	Lets you specify the mask channel name to save.

Example

```
.MaskChannelAdd "Mask1"
```

This example saves the current mask as a channel named Mask1 .

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskChannelDelete (PHOTO-PAINT)

.MaskChannelDelete .MaskID = *long*

This command deletes the specified mask channel.

Parameter	Description
.MaskID	Identifies the mask channel to delete. Channels for the active document are listed in the Channels Roll-Up. The channels are numbered and identified according to their position in the list ▸ the first listed channel is identified as 0, the second listed channel is identified as 1, and so on.

Example

```
.MaskChannelDelete 5
```

This example deletes the sixth mask channel.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskChannelLoad (PHOTO-PAINT)

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

This command loads a mask as a mask channel.

Parameter	Description
.FileName	Lets you specify the file name containing the mask to load.
.Left	Lets you specify the X-coordinate of the upper-left corner of the mask in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the mask in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the mask in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-left corner of the mask in pixels, relative to the origin.
.LoadType	Lets you specify the image loading method: 0 = All 1 = Partial 2 = Resample 3 = Crop

Example

```
.MaskChannelLoad "PIXELATE.CPT", 0, 0, 0, 0, 0
```

This example loads the mask channel named PIXELATE.CPT.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskChannelName (PHOTO-PAINT)

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

This command sets the name of an existing mask.

Parameter	Description
.MaskID	Identifies the mask channel. Channels for the active document are listed in the Channels Roll-Up. The channels are numbered and identified according to their position in the list ▸ the first listed channel is identified as 0, the second listed channel is identified as 1, and so on.
.Name	Lets you specify the new name of the mask channel.

Example

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

This example names the sixth mask channel.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskChannelSave (PHOTO-PAINT)

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

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

Parameter	Description
.MaskID	Identifies the mask channel. Channels for the active document are listed in the Channels Roll-Up. The channels are numbered and identified according to their position in the list ▸ the first listed channel is identified as 0, the second listed channel is identified as 1, and so on.

.FileName Lets you specify the name of the file.

.FilterID Lets you specify 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 Bitmap (JPG)
776 = Scitex CT Bitmap (SCT)
787 = GEM Paint File (IMG)
792 = OS/2 Bitmap (BMP)
1289 = Encapsulated PostScript (EPS)
1536 = Video for Windows (AVI)
1792 = Corel PHOTO-PAINT Image (CPT)
1794 = Corel CMX 5.0
1793 = Corel CMX 6.0
1795 = CorelDRAW (CDR)

.Compression	Lets you specify the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.MaskChannelSave 0, "MASK2.CPT", 1792, 0
```

This example saves the mask as MASK2.CPT.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskChannelToMask (PHOTO-PAINT)

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

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

Parameter	Description
.MaskID	Identifies the mask channel. Channels for the active document are listed in the Channels Roll-Up. The channels are numbered and identified according to their position in the list ▸ the first listed channel is identified as 0, the second listed channel is identified as 1, and so on.
.DrawMode	Lets you specify the draw mode

0 = Normal
1 = Add
2 = Subtract
3 = XOR

Example

```
.MaskChannelToMask 0, 0
```

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

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskToMaskChannel (PHOTO-PAINT)

.MaskToMaskChannel .MaskID = *long*

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

Parameter	Description
.MaskID	Identifies the mask channel. Channels the active document are listed in the Channels Roll-Up. The channels are numbered and identified according to their position in the list ▸ the first listed channel is identified as 0, the second listed channel is identified as 1, and so on.

Example

```
.MaskToMaskChannel 0
```

This example stores the active mask in the first mask channel.

{button ,AL(`OVR1 Mask channel commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Mask commands (PHOTO-PAINT)

MaskAffineDistort (PHOTO-PAINT)

.MaskAffineDistort .XOffset = *long*, .YOffset = *long*, .D11 = *double*, .D12 = *double*, .D21 = *double*, .D22 = *double*, .AntiAlias = *boolean*

This command is used by Corel SCRIPT to record manual manipulation of a mask area. If you want to distort a mask in Corel SCRIPT, use the MaskDistort command.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskAlign (PHOTO-PAINT)

.MaskAlign .Horizontal = *long*, .Vertical = *long*, .Center = *boolean*, .Grid = *boolean*, .ActObj = *boolean*, .SelObj = *boolean*, .Bkgnd = *boolean*

This command aligns the current mask on an image.

Parameter	Description
.Horizontal	Lets you specify the type of horizontal alignment: 0 = None 1 = Left 2 = Right 3 = Center
.Vertical	Lets you specify the type of vertical alignment: 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, to the document, to the selected object, or to the active object.
.Grid	Set to TRUE (-1) to align objects to the grid. Set to FALSE (0) to align objects to the center of the page, to the document, to the selected object, or to the active object.
.ActiveObject	Set to TRUE (-1) to align objects to the active object in the image. Set to FALSE (0) to align objects to the center of the page, to the grid, to the selected object, or to the document.
.SelectedObject	Set to TRUE (-1) to align selected objects to the document. Set to FALSE (0) to align objects to the center of the page, to the grid, to the selected object, or to the active object.
.Background	Set to TRUE (-1) to align selected objects to the background. Set to FALSE (0) to align objects to the center of the page, to the grid, to the selected object, or to the active object.

Example

```
.MaskAlign 3, 3, FALSE, FALSE, FALSE, FALSE, TRUE
```

This example aligns the mask marquee to the center of the document.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskBorder (PHOTO-PAINT)

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

This command creates a new mask border along the edges of the original mask.

Parameter	Description
.Width	Lets you specify the width of the mask border, in pixels.
.Edges	Lets you specify the type of feather: 0 = Soft 1 = Medium 2 = Hard

Example

```
.MaskBorder 10, 0
```

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

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskBrush (PHOTO-PAINT)

.MaskBrush .DrawMode = *long*, .Width = *long*, .Flatten = *long*, .Rotate = *long*, .NibShape = *long*, .Transparency = *long*, .SoftEdge = *long*

This command defines a mask using a brush stroke following a path defined by Draw commands. A MaskBrush command block must contain a series of StartDraw and ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 100%.
.Rotate	Lets you specify the rotation of the nib. Valid values range from 0 to 360 degrees.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Transparency	Lets you specify the transparency of the nib. Valid values range from 0 to 100%.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.MaskBrush 0, 8, 100, 0, 0, 0, 43, 0
    .StartDraw 16768, 11136, 0, 0
    .ContinueDraw 17082, 11540, 0, 0
     ...
    .ContinueDraw 49366, 54742, 0, 0
    .EndDraw
```

This example creates a mask outlined by the brush stroke defined by the StartDraw and ContinueDraw commands.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskCreate (PHOTO-PAINT)

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

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

Parameter	Description
.PreserveImage	Set to TRUE (-1) to preserve the image. Set to FALSE (0) to disable this option.
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR

Example

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

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

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskCreateFromPath (PHOTO-PAINT)

.MaskCreateFromPath .AntiAlias = *boolean*, .DrawMode = *long*

This command creates a mask selection that has the shape of the current path.

Parameter	Description
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR

Example

```
.MaskCreateFromPath -1, 1
```

This example creates a mask from an existing path using the Additive mask mode.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskDeFloat (PHOTO-PAINT)

.MaskDeFloat

This command is used after the MaskFloaterMoveTo command to draw the floating mask contents on the image and convert the mask to a non-floating mask.

Example

```
.MaskDeFloat
```

This example defloats the current mask.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskDistort (PHOTO-PAINT)

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

This command distorts the shape of the current mask.

Parameter	Description
.Corner1X	Lets you specify the X-coordinate of the new location of the upper-left corner of the mask's original bounding box in pixels, relative to the origin.
.Corner1Y	Lets you specify the Y-coordinate of the new location of the upper-left corner of the mask's original bounding box in pixels, relative to the origin.
.Corner2X	Lets you specify the X-coordinate of the new location of the upper-right corner of the mask's original bounding box in pixels, relative to the origin.
.Corner2Y	Lets you specify the Y-coordinate of the new location of the upper-right corner of the mask's original bounding box in pixels, relative to the origin.
.Corner3X	Lets you specify the X-coordinate of the new location of the lower-right corner of the mask's original bounding box in pixels, relative to the origin.
.Corner3Y	Lets you specify the Y-coordinate of the new location of the lower-right corner of the mask's original bounding box in pixels, relative to the origin.
.Corner4X	Lets you specify the X-coordinate of the new location of the lower-left corner of the mask's original bounding box in pixels, relative to the origin.
.Corner4Y	Lets you specify the Y-coordinate of the new location of the lower-left corner of the mask's original bounding box 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
```

This example distorts the selected mask by the specified parameters.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskEllipse (PHOTO-PAINT)

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

This command creates elliptical or circular masks.

Parameter	Description
.Left	Lets you specify the X-coordinate of the upper-left corner of the ellipse's mask in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the ellipse's mask in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the ellipse's mask in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the ellipse's mask in pixels, relative to the origin.
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Feather	Lets you specify 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
```

This example creates an elliptical mask with the given coordinates, with no feathering, and no anti-aliasing.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskExpand (PHOTO-PAINT)

.MaskExpand .Width = *long*

This command extends transparent areas of the mask into opaque areas by a given number of pixels.

Parameter	Description
.Width	Lets you specify the amount of stretching. Valid values range from 1 to 200 pixels.

Example

```
.MaskExpand 15
```

This example expands the mask by 15 pixels.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskFeather (PHOTO-PAINT)

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

This command feathers the edges of the mask, creating a smoothing effect.

Parameter	Description
.Width	Controls the number of pixels that are used in the feathering transition. A large number produces a wider transition area.
.Direction	Controls the direction of feathering 0 = Inside 1 = Middle 2 = Outside 3 = Average
.Type	Lets you specify the type of feather edge: 0 = Linear 1 = Curved

Example

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

This example uses 10 pixels in the feathering transition, an inside edge transition between a mask and the background, and a hard feather edge.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskFlipHorizontal (PHOTO-PAINT)

.MaskFlipHorizontal

This command flips the active mask horizontally.

Example

```
.MaskFlipHorizontal
```

This example flips the current mask horizontally.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskFlipVertical (PHOTO-PAINT)

.MaskFlipVertical

This command flips the active mask vertically.

Example

```
.MaskFlipVertical
```

This example flips the current mask vertically.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskFloaterMoveTo (PHOTO-PAINT)

.MaskFloaterMoveTo .Left = *long*, .Bottom = *long*, .Copy = *boolean*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 - *long*

This command moves the active mask and moves or copies the part of the image within the mask area.

Parameter	Description
.Left	Lets you specify the new X-coordinate of the lower-left corner of the mask's bounding box, in pixels.
.Bottom	Lets you specify the new Y-coordinate of the lower-left corner of the mask's bounding box, 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.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

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

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

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskFreehand (PHOTO-PAINT)

.MaskFreehand .DrawMode = *long*, .Feather = *long*, .AntiAlias = *long*

This command defines a mask using a path defined by Draw commands. A MaskFreehand command block must contain a series of StartDraw and ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Feather	Lets you specify the number of pixels you want to use along the mask selection's edge to apply feathering. Feathered pixels gradually become more opaque as you get closer to the protected area of the mask so that changes applied to the selection blend gradually towards the rest of the image. Valid values range from 0 to 100.
.AntiAlias	Set to 1 to apply anti-aliasing. Set to 0 to disable anti-aliasing.

Example

```
.MaskFreehand 0, 0, 1
    .StartDraw 166, 43, 0, 0
    .ContinueDraw 165, 43, 0, 0
     ...
    .ContinueDraw 221, 99, 0, 0
    .EndDraw
```

This example creates a mask defined by the StartDraw and ContinueDraw commands.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskGrow (PHOTO-PAINT)

.MaskGrow .AntiAlias = *boolean,* .MaskVisible = *boolean,* .ToleranceMode = *long,* .Normal = *long,* .Hue = *long,* .Saturation = *long,* .Brightness = *long*

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.

Parameter	Description
.Antialias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.MaskVisible	Set to TRUE (-1) to include all visible objects. Set to FALSE (0) to include the active object only.
.ToleranceMode	Lets you specify the tolerance mode: 0 = Normal 1 = HSB
.Normal	Lets you specify the tolerance as a percentage. This parameter is used only if .ToleranceMode is set to 0.
.Hue	Lets you specify the hue tolerance as a percentage. 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. This parameter is used only if .ToleranceMode is set to 1.
.Saturation	Lets you specify the saturation tolerance as a percentage. Saturation is the purity of a color. The more colors used to mix a color, the duller the color looks. This parameter is used only if .ToleranceMode is set to 1.
.Brightness	Lets you specify the brightness tolerance as a percentage. In the HSB color model, the component that determines the amount of black in a color. This parameter is used only if .ToleranceMode is set to 1.

Note

This command uses the current Magic Wand Tool tolerance settings.

Example

```
.MaskGrow TRUE, FALSE, 0,10,10,10,10
```

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

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskInvert (PHOTO-PAINT)

.MaskInvert

This command reverses the area included in the mask.

Example

```
.MaskInvert
```

This example inverts the current mask.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskLasso (PHOTO-PAINT)

.MaskLasso .DrawMode = *long*, .AntiAlias = *long*, .MaskVisible = *boolean*

This command defines a mask using a bounding area defined by Draw commands. The mask will snap to conform to the edges of the objects within the bounding area. A MaskLasso command block must contain a series of StartDraw and ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.AntiAlias	Set to 1 to apply anti-aliasing. Set to 0 to disable anti-aliasing.
.MaskVisible	Set to TRUE (-1) to include all visible objects in the selection. Set to FALSE (0) to include the active object only.

Example

```
.MaskLasso 0, TRUE, TRUE
    .ToleranceSettings 0, 10, 0, 0, 0
    .StartDraw 195, 156, 0, 0
    .ContinueDraw 194, 156, 0, 0
    ...
    .ContinueDraw 246, 217, 0, 0
    .EndDraw
```

This example creates a mask from the objects in the area defined by the StartDraw and ContinueDraw commands.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskLoad (PHOTO-PAINT)

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

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

Parameter	Description
.Filename	Lets you specify the name of the file.
.Left	Lets you specify the X-coordinate of the upper-left corner of the mask to be loaded in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the mask to be loaded in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the mask to be loaded in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the mask to be loaded in pixels, relative to the origin.
.LoadType	Lets you specify the image loading method: 0 = All 2 = Resample 3 = Crop

Note

A mask or image cannot be loaded as a partial file. The mask image is scaled to have the same dimensions as the current image.

Example

```
.MaskLoad "MASK1.CPT", 0, 0, 0, 0, 0
```

This example loads all of the mask named MASK1.CPT in to the upper-left corner of the image.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskMagicWand (PHOTO-PAINT)

.MaskMagicWand .ptX = *long*, .ptY = *long*, .DrawMode = *long*, .AntiAlias = *boolean*, .MaskVisible = *boolean,* .ToleranceMode = *long*, .Normal = *long*, .Hue = *long*, .Saturation = *long*, .Brightness = *long*

This command creates a contiguous mask that includes image pixels having colors within the given tolerance of the color at the starting position.

Parameter	Description
.ptX	Lets you specify the X-coordinate of the position at which the Magic Wand is applied, in pixels, relative to the origin.
.ptY	Lets you specify the Y-coordinate of the position at which the Magic Wand is applied, in pixels, relative to the origin.
.DrawMode	Lets you specify 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.
.MaskVisible	Set to TRUE (-1) to affect all visible objects. Set to FALSE (0) to affect the active object only.
.ToleranceMode	Lets you specify the tolerance mode: 0 = Normal. The .Normal parameter is used to set the tolerance level 1 = HSB The HSB parameters are used to set the tolerance level
.Normal	Lets you specify the Normal tolerance level. Valid values range from 0 to 100%.
.Hue	Lets you specify the Hue tolerance. 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. Valid values range from 0 to 100%.
.Saturation	Lets you specify the Saturation tolerance. Saturation is the purity of a color. The more colors used to mix a color, the duller the color looks. Valid values range from 0 to 100%.
.Brightness	Lets you specify the Brightness tolerance. In the HSB color model, the component that determines the amount of black in a color. Valid values range from 0 to 100%.

Example

```
.MaskMagicWand 285, 108, 0, TRUE, 0, 10, 10, 10, 10
```

This example creates a Magic Wand mask at the point (285, 108), applying anti-aliasing. The coordinates are expressed in pixels.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskPaint (PHOTO-PAINT)

.MaskPaint .Mask = *long*

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

Parameter	Description
.Mask	Lets you specify whether to paint on the mask or its image 1 = Paint on the mask 2 = Paint on the image

Example

```
.MaskPaint 1
```

This example paints on the mask.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskRectangle (PHOTO-PAINT)

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

This command defines rectangular masks.

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

Example

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

This 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(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskReduce (PHOTO-PAINT)

.MaskReduce .Width = *long*

This command extends opaque areas of the mask into transparent areas by a given number of pixels.

Parameter	Description
.Width	Lets you specify the amount by which the bounding box is inset, in pixels. Valid values range from 1 to 200.

Example

```
.MaskReduce 50
```

This example reduces the width of the mask by 50 pixels

```
.MaskReduce 60
```

This example reduces the width of the mask by 60 pixels.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskRemove (PHOTO-PAINT)

.MaskRemove

This command deletes the current mask .

Example

```
.MaskRemove
```

This example deletes the current mask.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskRemoveHoles (PHOTO-PAINT)

.MaskRemoveHoles

This command removes holes in the mask.

Example

```
.MaskRemoveHoles
```

This example removes holes created by masking tools.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskRotate (PHOTO-PAINT)

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

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

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

Example

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

This example rotates the mask 45 degrees around the center point located at X,Y coordinates 268, 363.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskSave (PHOTO-PAINT)

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

This command saves the active mask to a file.

Parameter	Description
.Filename	Lets you specify the path and name of the file.
.FilterID	Lets you specify 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 Bitmap (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	Lets you specify the type of file compression to apply: 0 = None 1 = LZW 2 = Packbits 3 = JPEG

Example

```
.MaskSave "MASK1.CPT", 1792, 0
```

This example saves the current mask as a file named MASK1.CPT.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskScissors (PHOTO-PAINT)

.MaskScissors .DrawMode = *long*, .Feather = *long*, .AntiAlias = *boolean,* .MaskVisible = *boolean*

This command defines a straight-edged mask using a bounding area defined by Draw commands. The mask will snap to approximately conform to the edges of the objects within the bounding area. A MaskScissors command block must contain a series of StartDraw and ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.Feather	Lets you specify the number of pixels you want to use along the mask selection's edge to apply feathering. Feathered pixels gradually become more opaque as you get closer to the protected area of the mask so that changes applied to the selection blend gradually towards the rest of the image. Valid values range from 0 to 100.
.Antialias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.MaskVisible	Set to TRUE (-1) to include all visible objects. Set to FALSE (0) to include the active object only.

Example

```
.MaskScissors 0, 0, FALSE, FALSE
    .StartDraw 78, 69, 0, 0
    .ContinueDraw 89, 76, 0, 0
    ...
    .ContinueDraw 87, 76, 0, 0
    .EndDraw
```

This example creates a scissors mask from the active object in the area defined by the StartDraw and ContinueDraw commands.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskSelectAll (PHOTO-PAINT)

.MaskSelectAll

This command creates a mask including the entire image area.

Example

```
.MaskSelectAll
```

This example encompasses the entire active image in a mask.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskSimilar (PHOTO-PAINT)

.MaskSimilar .AntiAlias = *boolean,* .MaskVisible = *boolean,* .ToleranceMode = *long,* .Normal = *long,* .Hue = *long,* .Saturation = *long,* .Brightness = *long*

This command adds to the current mask area any areas of the image having colors similar to the image color within the current mask area.

Parameter	Description
.Antialias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.MaskVisible	Set to TRUE (-1) to include all visible objects. Set to FALSE (0) to include the active object only.
.ToleranceMode	Lets you specify the tolerance mode: 0 = Normal 1 = HSB
.Normal	Lets you specify the tolerance as a percentage. This parameter is used only if .ToleranceMode is set to 0.
.Hue	Lets you specify the hue tolerance as a percentage. 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. This parameter is used only if .ToleranceMode is set to 1.
.Saturation	Lets you specify the saturation tolerance as a percentage. Saturation is the purity of a color. The more colors used to mix a color, the duller the color looks. This parameter is used only if .ToleranceMode is set to 1.
.Brightness	Lets you specify the brightness tolerance as a percentage. In the HSB color model, the component that determines the amount of black in a color. This parameter is used only if .ToleranceMode is set to 1.

Note

This command uses the current Magic Wand Tool tolerance settings.

Example

```
.MaskSimilar TRUE, FALSE, 0,10,10,10,10
```

This example expands the active mask to include all similarly masked colors.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskSkew (PHOTO-PAINT)

.MaskSkew .HorzAngle = *double*, .VertAngle = *double*, .AntiAlias = *boolean*

This command skews the current mask.

Parameter	Description
.HorzAngle	Lets you specify the horizontal skew angle in tenths of degrees. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew.
.VertAngle	Lets you specify 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.

Example

```
.MaskSkew 3.5, 11.5, TRUE
```

This example skews the mask by 3.5 degrees horizontally and 11.5 degrees vertically.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskSmooth (PHOTO-PAINT)

.MaskSmooth .Radius = *long*

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

Parameter	Description
.Radius	Lets you specify the radius of mask edge smoothing. Valid values range from 1 to 50.

Example

```
.MaskSmooth 5
```

This example applies a smoothing radius of 5 to the mask.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskStretch (PHOTO-PAINT)

.MaskStretch .Left = *long*, .Top = *long*, .XScale = *double*, .YScale = *double*, .AntiAlias = *boolean*

This command stretches the active mask.

Parameter	Description
.Left	Lets you specify the X-coordinate of the new upper-left corner of the stretched mask in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the new upper-left corner of the stretched mask in pixels, relative to the origin.
.XScale	Lets you specify the degree of horizontal stretch to apply to the mask.
.YScale	Lets you specify the degree of vertical stretch to apply to the mask.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.

Example

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

This example stretches the mask to the new coordinates shown.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskStroke (PHOTO-PAINT)

.MaskStroke .Mode = *long*

This command ends a BrushTool, ImageSprayerTool, or Eraser command block.

Parameter	Description
.Mode	Lets you specify the position of the brush stroke: 0 = Middle 1 = Inside 2 = Outside

Example

```
.BrushTool 1, 0, 0, 20, 0, 40, 75, 0, 100, 100
    .BrushTextureSettings "", 0, 0, 0, 0, 1, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 186, 159, 106, 0
    .MaskStroke 0
```

This example strokes the middle of a mask using the Brush tool.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskThreshold (PHOTO-PAINT)

.MaskThreshold .Level = *long*

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

Parameter	Description
.Level	Lets you specify the threshold level. Values greater than or equal to the threshold are set to white, values less than the threshold are set to black. Valid threshold levels range from 1 to 255.

Example

```
.MaskThreshold 128
```

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

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MaskTranslate (PHOTO-PAINT)

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

This command repositions the active mask.

Parameter	Description
.ptX	Lets you specify the number of pixels to horizontally translate the mask. A positive value moves to the right; a negative value moves to the left.
.ptY	Lets you specify the number of pixels to vertically translate the mask. A positive value moves down; a negative value moves up.

Example

```
.MaskTranslate 100, 14
```

This example translates the mask 100 pixels to the right and 14 pixels down.

{button ,AL(`OVR1 Mask commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Object commands (PHOTO-PAINT)

CreateBackground (PHOTO-PAINT)

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

This command creates a background in an image that has no background.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.CreateBackground 5, 0, 0, 0, 0
```

This example creates a black background.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndObjectTagWWWURL (PHOTO-PAINT)

.EndObjectTagWWWURL

This command ends an ObjectTagWWWURL command block.

Example

```
.ObjectTagWWWURL
    .ObjectURLInfo 1, 0, "www.object1.com", "Link to www.Object1.com"
    .ObjectURLInfo 2, 1, "www.sdfsdf.com", "Link to www.sdfsdf.com"
    .ObjectURLInfo 3, 2, "www.3.com", "Link to www.3.com"
    .EndObjectTagWWWURL
```

This example assigns URL tags to three objects in the current image. The first has a rectangular bounding box, the second, an oval bounding box, and the third, a polygonal bounding box.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectAddClipMask (PHOTO-PAINT)

.ObjectAddClipMask .Level = *long*

This command creates a clip mask that shows or hides objects in the Image Window.

Parameter	Description
.Level	Lets you specify the transparency of the clip mask. Set to 0 to hide the object in the Image Window. Set to 255 to show the object in the Image Window.

Example

```
.ObjectAddClipMask 255
```

This example adds a clip mask to all selected objects.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectAddClipMaskFromMask (PHOTO-PAINT)

.ObjectAddClipMaskFromMask .Inverted = *boolean*

This command creates a clip mask on the selected objects in an image. Creating a clip mask from a regular mask hides all pixels outside the selection.

Parameter	Description
.Inverted	Set to TRUE (-1) to invert the regular mask on the image before creating the clip mask. This allows you to create a clip mask from the area on the image that was originally masked or protected. Set to FALSE (0) to create a clip mask from the current selection.

Example

```
.ObjectAddClipMaskFromMask TRUE
```

This example inverts the mask and then creates a clip mask.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectAddClipMaskFromTransparency (PHOTO-PAINT)

.ObjectAddClipMaskFromTransparency

This command adds a clip mask to an object, using the object's shape. The clip mask is added to all selected objects.\

Example

```
.ObjectAddClipMaskFromTransparency
```

This example adds a clip mask to an object, based on the transparency values of the active object in an image.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectAffineDistort (PHOTO-PAINT)

.ObjectAffineDistort .XOffset = *long*, .YOffset = *long*, .D11 = *double*, .D12 = *double*, .D21 = *double*, .D22 = *double*, .AntiAlias = *boolean*

This command is used by Corel SCRIPT to record manual manipulation of an object. If you want to distort an object in Corel SCRIPT, use the ObjectDistort command.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

ObjectAlign (PHOTO-PAINT)

.ObjectAlign .Horizontal = *long*, .Vertical = *long*, .Center = *boolean*, .Grid = *boolean,* Active = *boolean,* SelToDoc = *boolean,* DistrBounds = *boolean,* HorizAlign = *boolean,* VertAlign = *boolean*

This command aligns selected objects to the grid, the page center, or to the topmost selected object.

Parameter	Description
.Horizontal	Lets you specify the type of horizontal alignment: 0 = None 1 = Left 2 = Right 3 = Center 4 = Width
.Vertical	Lets you specify the type of vertical alignment: 0 = None 1 = Top 2 = Bottom 3 = Center 4 = Height
.Center	Set to TRUE (-1) to align objects to the center of the page. Set to FALSE (0) to align objects to the grid, to the document, or to the active object.
.Grid	Set to TRUE (-1) to align objects to the grid. Set to FALSE (0) to align objects to the center of the page, to the document, or to the active object.
.Active	Set to TRUE (-1) to align objects to the active object in the image. Set to FALSE (0) to align objects to the center of the page, to the grid, or to the document.
.SelToDoc	Set to TRUE (-1) to align selected objects to the document. Set to FALSE (0) to align objects to the center of the page, to the grid, or to the active object.
.DistrBounds	Set to TRUE (-1) to distribute objects across the boundary of the image. Otherwise, set to FALSE (0).
.HorizAlign	Set to TRUE (-1) to align objects horizontally. Set to FALSE (0) to distribute objects horizontally.
.VertAlign	Set to TRUE (-1) to align objects vertically. Set to FALSE (0) to distribute objects vertically.

Example

```
.ObjectAlign 4, 2, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, TRUE
```

This example aligns objects along the bottom, distributing them horizontally by inter-object spacing (i.e., width) across the selected object's bounding box.

```
.ObjectAlign 3, 1, TRUE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE
```

This example aligns objects along the document's vertical, center axis, distributing them vertically according to their top edges across the selected object's bounding box.

```
.ObjectAlign 3, 1, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, TRUE
```

This example aligns objects along the top edge of the active object, distributing them horizontally according to their center points (across the entire document).

```
.ObjectAlign 2, 4, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, FALSE
```

This example aligns objects along the right edge of the document, distributing them vertically according to even inter-object spacing (i.e., height) across the selected object's bounding box.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectClip (PHOTO-PAINT)

.ObjectClip

This command clips selected objects to the current mask.

Example

```
.ObjectClip
```

This example clips the active object.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectClipToParent (PHOTO-PAINT)

.ObjectClipToParent .ObjectID = *long*, .Clip = *boolean*

This command clips the pixels of a child object to the shape of its parent object. An object is always the parent to the objects above it in the Objects Docker window.

Parameter	Description
.ObjectID	Lets you specify the index number of the object to be clipped.
.Clip	Set to TRUE (-1) to clip an object to its parent object. Otherwise, set to FALSE (0).

Example

```
.ObjectClipToParent 3 TRUE
```

This example clips the third object in an image to its parent object.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectColorTransparencyTool (PHOTO-PAINT)

.ObjectColorTransparencyTool .ToleranceMode = *long*, .ApplyToClipMask = *boolean*, .Smoothing = *long*

This command applies the color transparency to the active object.

Parameter	Description
.ToleranceMode	Lets you specify the tolerance mode. Normal HSB
.ApplyToClipMask	Set to TRUE (-1) to apply the transparency to the active object's clip mask. Otherwise, set to FALSE (0).
.Smoothing	Lets you specify a smoothing value for the transparency. Valid values range from 0 to 100.

Example

```
.ObjectColorTransparencyTool 0, TRUE, 27
    .ColorMaskColor 0, 5, 62, 155, 0, 0, 14, 30, 50, 10
    .ColorMaskColor 1, 5, 62, 155, 0, 0, 14, 30, 50, 10
    .ColorMaskColor 2, 5, 112, 102, 28, 0, 14, 30, 50, 10
    .EndObjectColorTransparencyTool
```

This example makes the selected colors transparent in the currently selected object using a clip mask.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectCombine (PHOTO-PAINT)

.ObjectCombine

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

Example

```
.ObjectCombine
```

This example combines the selected objects together.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectCreate (PHOTO-PAINT)

.ObjectCreate .PreserveImage = *boolean*

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

Parameter	Description
.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
```

This example creates an object from a mask.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectCreateFromBackground (PHOTO-PAINT)

.ObjectCreateFromBackground

This command creates an object from the image background. The background becomes an object in the image and is given an object name in the Objects Docker window. After you convert the background to an object, you can transform it with the transformation handles, move it, and edit it in a number of ways you were not able to before.

Example

```
.ObjectCreateFromBackground
```

This example creates an object from the image background.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectDefringe (PHOTO-PAINT)

.ObjectDefringe .Amount = *long*

This command spreads the interior colors of the selected objects out to their edges.

Parameter	Description
.Amount	Lets you specify the width of the Defringe effect, expressed in pixels. Valid values range from 1 to 100.

Example

```
.ObjectDefringe 10
```

This example applies a Defringe to the object with a width of 10 pixels.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectDelete (PHOTO-PAINT)

.ObjectDelete

This command deletes the selected object(s).

Example

```
.ObjectDelete
```

This example deletes the selected object(s).

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectDistort (PHOTO-PAINT)

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

This command distorts the shape of the selected object.

Parameter	Description
.Corner1X	Lets you specify the X-coordinate of the new location of the upper-left corner of the object's original bounding box in pixels, relative to the origin.
.Corner1Y	Lets you specify the Y-coordinate of the new location of the upper-left corner of the object's original bounding box in pixels, relative to the origin.
.Corner2X	Lets you specify the X-coordinate of the new location of the upper-right corner of the object's original bounding box in pixels, relative to the origin.
.Corner2Y	Lets you specify the Y-coordinate of the new location of the upper-right corner of the object's original bounding box in pixels, relative to the origin.
.Corner3X	Lets you specify the X-coordinate of the new location of the lower-right corner of the object's original bounding box in pixels, relative to the origin.
.Corner3Y	Lets you specify the Y-coordinate of the new location of the lower-right corner of the object's original bounding box in pixels, relative to the origin.
.Corner4X	Lets you specify the X-coordinate of the new location of the lower-left corner of the object's original bounding box in pixels, relative to the origin.
.Corner4Y	Lets you specify the Y-coordinate of the new location of the lower-left corner of the object's original bounding box 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
```

This example distorts the selected object by the specified parameters.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectDropShadow (PHOTO-PAINT)

.ObjectDropShadow .Mode = *boolean,* .Direction = *long*, .Distance = *long,* .Feather = *long*, .Type = *long*, .Edges = *long*, .Opacity = *long*, .Relative = *boolean,* .LightAngle = *long,* .Fade = *long,* .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Anchor = long, .Stretch = double

This command creates an object that has the same shape as the selected object and places it behind the original, producing a drop shadow effect. Optionally, you can use the Color parameters to alter the shadow's color.

Parameter	Description
.Mode	Set to TRUE (-1) to create a flat drop shadow. Set to FALSE (0) to add perspective to the drop shadow that you are creating.
.Direction	Lets you specify the position of the shadow relative to the original object. Values range from 0 to 360, where 0 is E and moving counterclockwise increases the angle.
.Distance	Lets you specify the horizontal and vertical distance between the edge of the original object and the outside edge of the shadow object.
.Feather	the width, in pixels, of the shadow's feathered edge. A feathered edge makes the shadow object blend gradually from its color to the image background. This makes the shadow object's edges less noticeable. Valid values range from 0 (sharpest) to 100.
.Type	Lets you specify the location of the feathered pixels, relative to the shadow object 0 = Inside. Places the feathered portion inside the shadow, along its edges 1 = Middle. Places approximately as many feathered pixels inside the edge as outside 2 = Outside. Adds pixels just outside the shadow's edges 3 = Average. Samples all pixels in the defined width and assigns a color value to each one individually. This results in some pixels being inside, some outside, and creates a more gradual transition in color between the shadow object and the background, much like a gradient.
.Edges	Lets you specify the edge type for the feathered portion of the shadow object 0 = Linear. Uses the sharp bends in the object to produce the feathered section 1 = Curved. Rounds the sharp edges.
.Opacity	Lets you specify the shadow object's opacity. Zero is completely transparent, 100 is completely opaque.
.Relative	Set to TRUE (-1) to measure the drop shadow's offset and feather width in percent. Set to FALSE (0) to measure the drop shadow's offset and feather width in the default units of measurement.
.LightAngle	Lets you specify the angle of the light source used to add perspective to the object's drop shadow. The angle of the light source is measured in degrees. Values range from 0 to 90.
Fade	Determines the percent by which the drop shadow fades as it extends away from the object.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.Anchor	Lets you specify where to anchor the drop shadow
.Stretch	Lets you specify the stretch value when you create a perspective shadow

Example

```
.ObjectDropShadow FALSE, 0, 23, 30, 3, 1, 100, TRUE, 0, 0, 5, 255, 51, 0, 0
```

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectDuplicate (PHOTO-PAINT)

.ObjectDuplicate

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.

Example

```
.ObjectDuplicate
```

This example creates a duplicate of the selected object.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectEdit (PHOTO-PAINT)

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

This command makes the specified object editable or uneditable.

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.
.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
```

This example makes the third object editable.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectEditAll (PHOTO-PAINT)

.ObjectEditAll

This command makes all objects editable, including the background.

Example

```
.ObjectEditAll
```

This example makes all objects editable including the background.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectEditNone (PHOTO-PAINT)

.ObjectEditNone

This command deselects the current selection.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectEditSelected (PHOTO-PAINT)

.ObjectEditSelected

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

Example

```
.ObjectEditSelected
```

This example makes all selected objects editable.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectEditTransparency (PHOTO-PAINT)

.ObjectEditTransparency .Mode = *long*

This command locks and unlocks an object's transparency.

Parameter	Description
.Mode	Set to 0 to lock the object's transparency. Set to 1 to unlock an object's transparency.

Example

```
.ObjectEditTransparency 0
```

This example locks object transparency.

```
.ObjectEditTransparency 1
```

This example unlocks object transparency.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectFeather (PHOTO-PAINT)

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

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

Parameter	Description
.Width	Lets you specify the number of pixels that are used in the feathering transition. Valid values range from 1 to 200.
.Type	Lets you specify 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 = Linear 1 = Curved

Example

```
.ObjectFeather 10, 0
```

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

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectFlipHorizontal (PHOTO-PAINT)

.ObjectFlipHorizontal

This command flips the selected object(s) horizontally.

Example

```
.ObjectFlipHorizontal
```

This example flips the selected object horizontally.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectFlipVertical (PHOTO-PAINT)

.ObjectFlipVertical

This command flips the selected object(s) vertically.

Example

```
.ObjectFlipVertical
```

This example flips the selected object vertically.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectGroup (PHOTO-PAINT)

.ObjectGroup

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

Example

```
.ObjectGroup
```

This example groups selected objects together.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectMerge (PHOTO-PAINT)

.ObjectMerge .All = *boolean*

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

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

Example

```
.ObjectMerge -1
```

This example merges all objects.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectMergeMode (PHOTO-PAINT)

.ObjectMergeMode .MergeMode = *long*

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

Parameter	Description
.MergeMode	Lets you specify 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 that can be used depend on the image type.

Example

```
.ObjectMergeMode 11
```

This example sets the object merge mode to Saturation.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectName (PHOTO-PAINT)

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

This command sets the name of the specified object.

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.
.Name	Lets you specify the new name of the object.

Example

```
.ObjectName 1, "Capricorn"
```

This example names the second object "Capricorn".

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectNew (PHOTO-PAINT)

.ObjectNew .Name = *string,* .Opacity = *long,* .MergeMode = *long*

This command creates a new object.

Parameter	Description	
.Name	Lets you specify the name of the object you are creating.	
.Opacity	Lets you specify the object's opacity. Valid values range from 0 to 100.	
.MergeMode	.MergeMode	Lets you specify the Merge Mode:
	0 = Normal	12 = Lum
	1 = Add	13 = Invert
	2 = Subtract	14 = And
	3 = Difference	15 = Or
	4 = Multiply	16 = Xor
	5 = Divide	17 = Red
	6 = Lighter	18 = Green
	7 = Darker	19 = Blue
	8 = Texturize	20 = Cyan
	9 = Color	21 = Magenta
	10 = Hue	22 = Yellow
	11 = Saturation	23 = Black

Example

```
ObjectNew "Object1", 100, 0
```

This example creates an object with 100% opacity using the Normal merge mode.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectOpacity (PHOTO-PAINT)

.ObjectOpacity .Amount = *long*

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

Parameter	Description
.Amount	Lets you specify the opacity. Valid values range from 0 to 99 percent.

Example

```
.ObjectOpacity 50
```

This example sets the object's opacity to 50%.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectOrder (PHOTO-PAINT)

.ObjectOrder .Order = *long*

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

Parameter	Description
.Order	Lets you specify 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 0
```

This example orders the selected object to the front of the document

```
.ObjectOrder 3
```

This example orders the selected object back one.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectOrderChange (PHOTO-PAINT)

.ObjectOrderChange .Source = *long*, .Destination = *long*

This command alters the order of the objects designated by the Source and Destination IDs.

Parameter	Description
.Source	The Object ID of the object to be moved.
.Destination	The final position of the Source object.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectProperties (PHOTO-PAINT)

.ObjectProperties .ObjectID = *long*, .Name = *string*, .Opacity = *long*, .MergeMode = *long*, .Visible = *boolean*, .Clip = *boolean*, .Enable = *boolean*, .Link = *boolean*

This command changes an object's properties.

Parameter	Description
.ObjectID	Lets you specify the index number of the object.
.Name	Lets you specify the name of the object.
.Opacity	Lets you specify the opacity of the object.
.MergeMode	Lets you specify 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
.Visible	Set to TRUE (-1) to make the object visible. Set to FALSE (0) to make the object invisible.
.Clip	Set to TRUE (-1) to clip the object to its parent object. Otherwise, set to FALSE (0).
.Enable	Set to TRUE (-1) to enable the object's clip mask. Set to FALSE (0) to disable the object's clip mask.
.Link	Set to TRUE (-1) to link the object to its clip mask. Otherwise, set to FALSE (0).

Example

```
.ObjectProperties 1, "Object 1", 100, 0, TRUE, FALSE, FALSE, FALSE
```

This example changes the properties of an object named "Object 1.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectRemoveClipMask (PHOTO-PAINT)

.ObjectRemoveClipMask .Apply = *boolean*

This command removes a clip mask from the selected objects.

Parameter	Description
.Apply	Set to TRUE (-1) to combine the clip mask with the object before removing it. Set to FALSE (0) to remove the clip mask without applying it to the object.

Example

```
.ObjectRemoveClipMask FALSE
```

This example removes the clip mask without applying it.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectRemoveMatte (PHOTO-PAINT)

.ObjectRemoveMatte .White = *boolean*

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.

Parameter	Description
.White	Set to TRUE (-1) to remove white matting. Set to FALSE (0) to remove black matting.

Example

```
.ObjectRemoveMatte -1
```

This example removes the black or white matting from an object.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectRotate (PHOTO-PAINT)

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

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

Parameter	Description
.XCenter	Lets you specify the X-coordinate of the center of rotation in pixels, relative to the origin.
.YCenter	Lets you specify the Y-coordinate of the center of rotation in pixels, relative to the origin.
.Angle	Lets you specify 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 = -45.0
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.Copy	Set to TRUE (-1) to apply the rotate command to a copy of the object. Set to FALSE (0) to apply the command to the original object without making a copy.

Example

```
.ObjectRotate 180, 252, -45.0, -1
```

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

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectSelect (PHOTO-PAINT)

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

This command selects or deselects the specified object.

Parameter	Description
.ObjectID	Lets you specify the object. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.
.Selected	Set to TRUE (-1) to select the object. Set to FALSE (0) to deselect the object.

▸ **Note**

Selecting an object makes it visible and editable. Selecting or deselecting an object in a group selects or deselects all objects in the group.

Example

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

This example selects object number 2.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectSelectAll (PHOTO-PAINT)

.ObjectSelectAll

This command selects all objects.

Example

```
.ObjectSelectAll
```

This example selects all objects.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectSelectNone (PHOTO-PAINT)

.ObjectSelectNone

This command deselects any selected objects.

Example

```
.ObjectSelectNone
```

This example deselects all selected objects.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectSkew (PHOTO-PAINT)

.ObjectSkew .HorzAngle = *double*, .VertAngle = *double*, .AntiAlias = *boolean*, .Copy = *boolean*

This command skews the selected object(s).

Parameter	Description
.HorzAngle	Lets you specify the horizontal skew angle in tenths of degrees. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew.
.VertAngle	Lets you specify 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.
.Copy	Set to TRUE (-1) to apply the skew command to a copy of the object. Set to FALSE (0) to apply the command to the original object without making a copy.

Example

```
.ObjectSkew -300, -300, TRUE, TRUE
```

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

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectStretch (PHOTO-PAINT)

.ObjectStretch .Left = *long*, .Top = *long*, .XScale = *double*, .YScale = *double*, .AntiAlias = *boolean*, .Copy = *boolean*

This command stretches the selected object(s).

Parameter	Description
.Left	Lets you specify the X-coordinate of the new upper-left corner of the object's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the new upper-left corner of the object's bounding box in pixels, relative to the origin.
.XScale	Lets you specify the degree of horizontal stretch to apply to the object.
.YScale	Lets you specify the degree of vertical stretch to apply to the object.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.Copy	Set to TRUE (-1) to apply the stretch command to a copy of the object. Set to FALSE (0) to apply the command to the original object without making a copy.

Example

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

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

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectTagWWWURL (PHOTO-PAINT)

.ObjectTagWWWURL

This command assigns a URL tag to objects in your image. An ObjectTagWWWURL command block must contain one or more ObjectURLInfo commands, and end with an EndObjectTagWWWURL command.

Example

```
.ObjectTagWWWURL
    .ObjectURLInfo 1, 0, "www.object1.com", "Link to www.Object1.com"
    .ObjectURLInfo 2, 1, "www.sdfsdf.com", "Link to www.sdfsdf.com"
    .ObjectURLInfo 3, 2, "www.3.com", "Link to www.3.com"
    .EndObjectTagWWWURL
```

This example assigns URL tags to three objects in the current image. The first has a rectangular bounding box, the second, an oval bounding box, and the third, a polygonal bounding box.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectThreshold (PHOTO-PAINT)

.ObjectThreshold .Threshold = *long*

This command alters the edges of mask selections or objects, and is accessible from both the Mask and Object menus. It is used to remove the smooth transition between a selection, or an object, and the underlying image. Such transitions are created by processes such as feathering. Setting a threshold value results in crisp and obvious edges for the current mask selection or selected object.

The object or mask marquee will be placed along the pixels in the feathered portion that have the grayscale value specified. The grayscale value of the pixels located on either side of the marquee are assigned a value of 0 or 255. This makes them part of the selection or object, or excludes them. A low threshold value includes more pixels in the selection or object than a high value.

Parameter	Description
.Threshold	Lets you specify the grayscale value of the pixels you want the mask selection edge, or object edge, to be located on. Valid values range from 0 (black) to 255 (white).

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectToggleClipMask (PHOTO-PAINT)

.ObjectToggleClipMask .Clip = *boolean*

This command enables or disables a clip mask.

Parameter	Description
.Clip	Set to TRUE (-1) to enable the clip mask. Set to FALSE (0) to disable the clip mask again.

Example

```
.ObjectToggleClipMask FALSE
```

This example disables the clip mask.

```
.ObjectToggleClipMask TRUE
```

This example enables the clip mask.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectToggleLinkClipMask (PHOTO-PAINT)

.ObjectToggleLinkClipMask .ObjectID = *long*, .Link = *boolean*

This command specifies whether the object and its associated clip mask are linked together.

Parameter	Description
.ObjectID	Lets you specify the object to which the clip mask will be linked.
.Link	Set to TRUE (-1) to link the object and its associated clip mask. Otherwise, set to FALSE (0).

Example

```
.ObjectToggleLinkClipMask 3 TRUE
```

This example links the object and its associated clip mask.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectTranslate (PHOTO-PAINT)

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

This command repositions selected object(s).

Parameter	Description
.ptX	Lets you specify the number of pixels to horizontally translate the object(s) or their copies. A positive value moves or copies to the right; a negative value moves or copies to the left.
.ptY	Lets you specify the number of pixels to vertically translate the object(s) or their copies. A positive value moves or copies downwards; a negative value moves or copies upwards.
.Copy	Set to TRUE (-1) to copy the object(s). Set to FALSE (0) to translate the object(s).

Note

If the objects are copied, the copies remain selected and the originals are de-selected when the command complete execution.

Example

```
.ObjectTranslate 15, -10, 0
```

This example moves the selected object(s) 15 pixels to the right and 10 pixels up.

```
.ObjectTranslate 103, 25, -1
```

This example copies the selected object(s), and places the copied object(s) 103 pixels to the right of and 25 pixels below the selected object(s)'s position.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectTransparencyTool (PHOTO-PAINT)

.ObjectTransparencyTool .Start = *long*, .End = *long*, .UseOriginal = *boolean,* .Handles = *long,* .ApplyToClipMask = *boolean*

This command changes the transparency of objects in your image.

Parameter	Description
.Start	Lets you specify the transparency value for the start point of the object's transparency blend with the rest of the image. Valid values range from 0 to 100%.
.End	Lets you specify the transparency value for the end point of the object's transparency blend with the rest of the image. Valid values range from 0 to 100%.
.UseOriginal	Set to TRUE (-1) to add the transparency value you set to the transparency value of the pixels in the object. Set to FALSE (0) to replace the existing transparency values of the object's pixels with the values set for this tool.
.Handles	Lets you specify the number of handles used to apply the transparency.
.ApplyToClipMask	Set to TRUE (-1) to apply the transparency to the object's clip mask. Otherwise, set to FALSE (0).

Example

```
.ObjectTransparencyTool 0, 100, TRUE, 4, FALSE
    .GradientPoint 0, -17, 112, 9, 0, 0, 0, 0, 255
    .GradientPoint 1, -22, 113, 9, 167, 0, 0, 0, 88
    .GradientPoint 2, -28, 115, 9, 76, 0, 0, 0, 179
    .GradientPoint 3, -33, 115, 9, 255, 0, 0, 0, 0
    .Gradient 2, 50
```

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectUngroup (PHOTO-PAINT)

.ObjectUngroup

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

Example

```
.ObjectUngroup
```

This example reverses the ObjectGroup command, and ungroups the grouped objects.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectURLInfo (PHOTO-PAINT)

.ObjectURLInfo .ObjectID = *long*, .Region = *long*, .Address = *string*, .Comments = *string*

This command sets the URL attributes for the ObjectTagWWWURL command. This command must appear inside an ObjectTagWWWURL command block.

Parameter	Description
.ObjectID	Lets you specify the index number of the object to be tagged.
.Region	Lets you specify the type of bounding region: 0 = Rectangular 1 = Oval 2 = Polygonal 3 = Circular
.Address	Lets you specify the address string of the URL.
.Comments	Lets you specify the comments that appear on the command line when you place your cursor over the URL bounding region.

Example

```
.ObjectTagWWWURL
    .ObjectURLInfo 1, 0, "www.object1.com", "Link to www.Object1.com"
    .ObjectURLInfo 2, 1, "www.sdfsdf.com", "Link to www.sdfsdf.com"
    .ObjectURLInfo 3, 2, "www.3.com", "Link to www.3.com"
    .EndObjectTagWWWURL
```

This example assigns URL tags to three objects in the current image. The first has a rectangular bounding box, the second, an oval bounding box, and the third, a polygonal bounding box.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ObjectVisible (PHOTO-PAINT)

.ObjectVisible .StartIndex = *long*, .EndIndex = *long,* .Visible = *boolean*

This command hides or shows the specified object.

Parameter	Description
.StartIndex	Lets you specify the first object in the range. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.
.EndIndex	Lets you specify the last object in the range. Objects are numbered and identified according to their image order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.
.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 1, 4, TRUE
```

This example makes objects 1 to 4 visible.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

.LensNew (PHOTO-PAINT)

.LensNew .Name=*string*

This command creates a new lens object that covers the entire image and to which an ImageEffect command is applied. All parameters assigned to the ImageEffect command are applied to the lens object instead of being applied to the image pixels themselves.

Parameter	Description
.Name	Determines the name to assign to the lens. This name is displayed in the Objects Roll-Up next to the lens's thumbnail.

Example

```
.ImageBCI 13, -22, 11
    .LensNew "B-C-I Lens 1"
    .EndColorEffect
```

This example creates a Brightness/Contrast/Intensity lens that covers the entire image and that has the name "B-C-I Lens 1" displayed in the Objects Roll-Up.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

.LensCreateFromMask (PHOTO-PAINT)

.LensCreateFromMask .Name=*string*

This command creates a new lens by converting the mask selection displayed in the Image Window into a lens. This command must be preceeded by an ImageEffect command. All parameters assigned to the ImageEffect command are applied to the lens object instead of being applied to the image pixels themselves.

Parameter	Description
.Name	Determines the name to assign to the lens. This name is displayed in the Objects Roll-Up next to the lens's thumbnail.

Example

```
.ImageHSL 20, -20, 30
    .LensCreateFromMask "HSL Lens 2"
    .EndColorEffect
```

This example converts the current mask selection to a Hue/Saturation/Lightness lens "HSL Lens 2" displayed in the Objects Roll-Up.

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

.LensEdit (PHOTO-PAINT)

.LensEdit .ObjectID=*long,* Name=*string*

This command changes the properties of an existing lens. You can also use this command to change the ImageEffect command used in the lens. This command must be preceeded by an ImageEffect command which determines the effect and properties to assign to the existing lens you decide to edit.

Parameter	Description
.ObjectID	Lets you specify the lens object. All objects are numbered and identified according to their position in the stacking order (from back to front) ▸ the first object is identified as 0, the second from the back is identified as 1, and so on. The background is considered an object, and because no other object can be placed below the background, it is always identified as 0.

.Name Determines the name to assign to the lens. This name is displayed in the Objects Roll-Up next to the lens's thumbnail. You can keep the current name of the lens or change it as you change its properties.

Example

```
.ImageLevelThreshold 0, 179,255, 255, 0
    .LensEdit 1, "Threshold Lens 3"
    .EndColorEffect
```

This example modifies the lens that is just above the image background and makes that lens use the ImageLevelThreshold command and changes the lens's name to "Threshold Lens 3".

{button ,AL(`OVR1 Object commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Movie commands (PHOTO-PAINT)

MovieBackOne (PHOTO-PAINT)

.MovieBackOne

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

Example

```
.MovieBackOne
```

This example moves the movie back by one frame.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieCreate (PHOTO-PAINT)

.MovieCreate

This command creates a new movie document using the current image.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MovieDeleteFrame (PHOTO-PAINT)

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

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

Parameter	Description
.FromFrame	Lets you specify the first frame to be deleted from the movie.
.ToFrame	Lets you specify the last frame to be deleted from the movie.

Example

```
.MovieDeleteFrame 7, 9
```

This example deletes movie frames number 7, 8, and 9.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieForward (PHOTO-PAINT)

.MovieForward

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

Example

```
.MovieForward
```

This example advances the movie to the last frame.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieForwardOne (PHOTO-PAINT)

.MovieForwardOne

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

Example

```
.MovieForwardOne
```

This example advances the movie by one frame.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieFrameDelay (PHOTO-PAINT)

.MovieFrameDelay .StartFrame = *long*, .EndFrame = *long*, .Delay = *long*

This command specifies the delay, in milliseconds from the first frame in a movie to the last.

Parameter	Description
.StartFrame	Lets you specify the first frame the sequence.
.EndFrame	Lets you specify the last frame in the sequence.
.Delay	Lets you specify the delay in milliseconds.

Example

```
.MovieFrameRate 25
.MovieFrameDelay 1, 5, 200
.MovieFrameDelay 6, 10, 300
.MovieFrameDelay 11, 25, 200
.EndMovieFrameRate
```

This example sets the frame rate for each frame in the movie. Frames 1 to 5 are set to 200 milliseconds. Frames 6 to 10 are set to 300 milliseconds. Frames 11 to 25 are set to 200 milliseconds.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MovieFrameRate (PHOTO-PAINT)

.MovieFrameRate .Frames = *long*

This command sets the frame delay, in milliseconds, for each specified movie frame. Frame rates are run-length encoded, which means that you can specify the delay for each frame or a range of frames in the movie.

Parameter	Description
.Frames	Lets you specify the number of frames to change.

Example

```
.MovieFrameRate 25
.MovieFrameDelay 1, 5, 200
.MovieFrameDelay 6, 10, 300
.MovieFrameDelay 11, 25, 200
.EndMovieFrameRate
```

This example sets the frame rate for each frame in the movie. Frames 1 to 5 are set to 200 milliseconds. Frames 6 to 10 are set to 300 milliseconds. Frames 11 to 25 are set to 200 milliseconds.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndMovieFrameRate (PHOTO-PAINT)

.EndMovieFrameRate

This command ends the movie frame rate sequence.

Example

```
.MovieFrameRate 25
.MovieFrameDelay 1, 5, 200
.MovieFrameDelay 6, 10, 300
.MovieFrameDelay 11, 25, 200
.EndMovieFrameRate
```

This example sets the frame rate for each frame in the movie. Frames 1 to 5 are set to 200 milliseconds. Frames 6 to 10 are set to 300 milliseconds. Frames 11 to 25 are set to 200 milliseconds.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieGotoFrame (PHOTO-PAINT)

.MovieGotoFrame .Frame = *long*

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

Parameter	Description
.Frame	Lets you specify the frame number to which you want to advance.

Example

```
.MovieGotoFrame 5
```

This example makes movie frame number 5 the active image

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MovieInsertFile (PHOTO-PAINT)

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

This command inserts any image file into the active movie file.

Parameter	Description
.FileName	Lets you specify the name of the file to be inserted.
.Left	Lets you specify the X-coordinate of the upper-left corner of the file to be inserted in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the file to be inserted in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the file to be inserted in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the file to be inserted in pixels, relative to the origin.
.LoadType	Lets you specify the image loading method: 0 = All 2 = Resample 3 = Crop
.StartFrame	Lets you specify 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.

Note

The loaded image will be scaled to the movie frame dimensions.

Example

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

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

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieInsertFrame (PHOTO-PAINT)

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

This command inserts one or more frames into a movie.

Parameter	Description
.StartFrame	Lets you specify the insertion point of a frame or series of frames.
.NumberOfFrames	Lets you specify 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, 6, 0, 0
```

This example inserts 6 movie frames after movie frame number 5, using the paper color

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

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

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MovieMoveFrame (PHOTO-PAINT)

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

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

Parameter	Description
.FromFrame	Lets you specify the first frame included in the frame move.
.ToFrame	Lets you specify the last frame included in the frame move.
.MoveToFrame	Lets you specify 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
```

This example moves frames number 5 through 12 and repositions them after frame number 1

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

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

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MovieRewind (PHOTO-PAINT)

.MovieRewind

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

Example

```
.MovieRewind
```

This example rewinds the movie back to the beginning.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

MovieSelectPartial (PHOTO-PAINT)

.MovieSelectPartial .StartFrame = *long*, .EndFrame = *long*

This command loads selected frames from a movie file.

Parameter	Description
.StartFrame	Lets you specify the first frame to load.
.EndFrame	Lets you specify the last frame in the sequence.

{button ,AL(`OVR1 Movie commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Drawing commands (PHOTO-PAINT)

ContinueDraw (PHOTO-PAINT)

.ContinueDraw .ptX = *long*, .ptY = *long*, .Timer = *long*, .Pressure = *long,* .Tilt = *long,* .Rotate = *long*

This command continues drawing operations started with the .StartDraw command such as drawing, masking, and erasing.

Parameter	Description
.ptX	Lets you specify the X-coordinate of the point to continue drawing. For Brush, Effects and Clone tools, this value is expressed in 1/256 of a pixel, relative to the origin. For other tools, this value is expressed in pixels, relative to the origin.
.ptY	Lets you specify the Y-coordinate of the point to continue drawing. For Brush, Effects and Clone tool, this value is expressed in 1/256 of a pixel, relative to the origin. For other tools, this value is expressed in pixels, relative to the origin.
.Timer	Used for tools like the airbrush, where the nib size changes with time. Values start at 0 and increment by 1 for every clock cycle.
.Pressure	Used for a Pen Tablet that has pressure sensitivity. Values range from 0 to 9999.
.Tilt	Lets you specify the tilt attribute of the clone brush tool. Values range from 0-90.
.Rotate	Lets you specify the rotation angle of the clone brush tool. Values range from 0-360.

Example

```
.ImageSprayerTool "fruit.cpt", 1, 5, 87, 0, 0
    .ImageSprayerSettings 1, 11, 19, 0, 0, 1, 5, 1
    .RandomSeed 552585248
    .StartDraw 11904, 8320, 0, 0
    .ContinueDraw 14343, 9096, 0, 0
    ...
    .ContinueDraw 61761, 18445, 0, 0
    .EndDraw
```

This example draws a brush stroke containing a variety of fruits.

```
.Eraser 20, 100, 0, 0, 0, 0, 0
    .SetPaperColor 5, 255, 255, 255, 0
    .StartDraw 40320, 8832, 0, 0
    .ContinueDraw 39040, 8832, 0, 0
    ...
    .ContinueDraw 59678, 18166, 0, 0
    .EndDraw
```

This example erases a portion of your image defined by the draw path.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Ellipse (PHOTO-PAINT)

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

This command draws ellipses.

Parameter	Description
.Left	Lets you specify the X-coordinate of the upper-left corner of the ellipse's bounding box in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the ellipse's bounding box in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the ellipse's bounding box in pixels, relative to the origin.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the ellipse's bounding box in pixels, relative to the origin.

Note

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

Example

```
.EllipseTool 0, 0, 0, TRUE, TRUE, TRUE
    .FillBitmap "d:\Fill.cpt", 254000, 254000, 0, 0, FALSE, 0, FALSE, FALSE, TRUE
    .Ellipse 51, 199, 232, 356
```

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

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndDraw (PHOTO-PAINT)

.EndDraw

This command ends a sequence of drawing commands used with drawing, masking, and erasing tools.

Example

```
.ImageSprayerTool "fruit.cpt", 1, 5, 87, 0, 0
    .ImageSprayerSettings 1, 11, 19, 0, 0, 1, 5, 1
    .RandomSeed 552585248
    .StartDraw 11904, 8320, 0, 0
    .ContinueDraw 14343, 9096, 0, 0
    ...
    .ContinueDraw 61761, 18445, 0, 0
    .EndDraw
```

This example draws a brush stroke containing a variety of fruits.

```
.Eraser 20, 100, 0, 0, 0, 0, 0
    .SetPaperColor 5, 255, 255, 255, 0
    .StartDraw 40320, 8832, 0, 0
    .ContinueDraw 39040, 8832, 0, 0
    ...
    .ContinueDraw 59678, 18166, 0, 0
    .EndDraw
```

This example erases a portion of your image defined by the draw path..

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Eraser (PHOTO-PAINT)

.Eraser .Width = *long*, .Flatten = *long*, .Rotate = *long*, .NibShape = *long*, .Transparency = *long*, .SoftEdge = *long*

This command erases a portion of your image defined by a series of Draw commands. An Eraser command block must contain a series of StartDraw and ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 100%.
.Rotate	Lets you specify the rotation of the nib. Valid values range from 0 to 360 degrees.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Transparency	Lets you specify the transparency of the nib. Valid values range from 0 to 100%.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.Eraser 20, 100, 0, 0, 0, 0
    .SetPaperColor 5, 255, 255, 255, 0
    .StartDraw 40320, 8832, 0, 0
    .ContinueDraw 39040, 8832, 0, 0
    ...
    .ContinueDraw 59678, 18166, 0, 0
    .EndDraw
```

This example replaces the portion of your image defined by the Draw commands with the defined paper color.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Fill (PHOTO-PAINT)

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

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

Parameter	Description
.Left	Lets you specify the X-coordinate of the point where filling begins in pixels, relative to the origin.
.Top	Lets you specify 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:
.FillTool
.ToleranceSettings

The .Fill command must be preceded by appropriate setup commands from the following group:
.FillSolid
.FillFountainColor and .FillFountainApply
.FillBitmap
.FillTexture

Example

```
FillTool 0, 0
    .ToleranceSettings 0, 100, 10, 10, 10
    .FillSolid 5, 0, 0, 255, 0
    .Fill 73, 53, TRUE
```

This example applies a solid blue fill beginning at the point (73, 53) with no anti-aliasing. These coordinates are expressed in pixels.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillFountain (PHOTO-PAINT)

.FillFountain

This command creates a fountain fill in the current image.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Gradient (PHOTO-PAINT)

.Gradient .Type = *long*, .MarkerPos = *long*

This command creates a gradient fill in the current image, ending a GradientTool command block.

Parameter	Description
.Type	Lets you specify the type of gradient: 1 = Flat 2 = Linear 3 = Elliptical 4 = Radial 5 = Rectangular 6 = Square 7 = Conical 8 = Bitmap 9 = Texture
.MarkerPos	Marks the halfway point of the transparency range in the color gradient. Drag the slider to move the halfway point of the transparency range to a new position on the gradient.

Example

```
.GradientTool 5, 0, 0, 4
    .SetPaintColor 5, 102, 255, 0, 0
    .SetPaperColor 8, 0, 0, 0, 0
    .GradientPoint 0, 23, 169, 5, 255, 255, 255, 0, 255
    .GradientPoint 1, 65, 133, 5, 153, 255, 0, 0, 255
    .GradientPoint 2, 124, 82, 5, 255, 51, 0, 0, 255
    .GradientPoint 3, 173, 39, 8, 0, 0, 0, 0, 255
    .Gradient 2, 50
```

This example applies a gradient fill to the active objects.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

GradientPoint (PHOTO-PAINT)

.GradientPoint .Index = *long*, .PtX = *long*, .PtY = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Transparency = *long*

This command sets the gradient point for an image.

Parameter	Description
.Index	References the nodes created on the gradient. The start node is always 0.
.PtX	Lets you specify the horizontal coordinate (x) of the gradient.
.PtY	Lets you specify the vertcial coordinate (x) of the gradient.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.Transparency	Lets you specify the transparency of each node applied to the gradient. Valid values range from 0 to 100.

Example

```
.GradientTool 5, 0, 0, 4
    .SetPaintColor 5, 102, 255, 0, 0
    .SetPaperColor 8, 0, 0, 0, 0
    .GradientPoint 0, 23, 169, 5, 255, 255, 255, 0, 255
    .GradientPoint 1, 65, 133, 5, 153, 255, 0, 0, 255
    .GradientPoint 2, 124, 82, 5, 255, 51, 0, 0, 255
    .GradientPoint 3, 173, 39, 8, 0, 0, 0, 0, 255
    .Gradient 2, 50
```

This example applies a gradient fill to the active objects.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Rectangle (PHOTO-PAINT)

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

This command draws rectangles and rounded rectangles.

Parameter	Description
.Left	Lets you specify the X-coordinate of the upper-left corner of the rectangle in pixels, relative to the origin.
.Top	Lets you specify the Y-coordinate of the upper-left corner of the rectangle in pixels, relative to the origin.
.Right	Lets you specify the X-coordinate of the lower-right corner of the rectangle in pixels, relative to the origin.
.Bottom	Lets you specify 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:

.RectangleTool
.FillSolid
.FillBitmap
.FillTexture

Example

```
.RectangleTool 0, 0, 0, 0, -1, -1, -1
    .FillSolid 3, 0, 0, 255, 0
    .Rectangle 20, 20, 200, 200
```

This example creates a solid blue rectangle.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

StartCloneDraw (PHOTO-PAINT)

.StartCloneDraw .SrcPtX = *long*, .SrcPtY = *long*, .DestPtX = *long*, .DestPtY = *long,* .Tilt = *long,* .Rotate = *long*

This command begins a stroke of the clone tool. A StartCloneDraw command block must contain one or more ContinueDraw commands and end with an EndDraw command.

Parameter	Description
.SrcPtX	Lets you specify the horizontal coordinate of the point in your drawing you want to clone. Expressed in normal coordinates.
.SrcPtY	Lets you specify the vertical coordinate of the point in your drawing you want to clone. Expressed in normal coordinates.
.DestPtX	Lets you specify the horizontal coordinate of the point you want to clone to. Expressed in 1/256 of a pixel.
.DestPtY	Lets you specify the vertical coordinate of the point you want to clone to. Expressed in 1/256 of a pixel.
.Tilt	Lets you specify the tilt attribute of the clone brush tool. Values range from 0-90.
.Rotate	Lets you specify the rotation angle of the clone brush tool. Values range from 0-360.

Example

```
.CloneTool 0, 1, 0, 20, 0, 25, 0, 0, 100, 25
    .BrushTextureSettings "", 0, 0, 0, 0, 10, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 255, 255, 0, 0
    .RandomSeed 1905306265
    .StartCloneDraw 235, 100, 73528, 30890, 0, 0
    .ContinueDraw 71846, 30273, 0, 0
    .....
    .ContinueDraw 71408, 29783, 0, 0
    .EndDraw
```

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

StartDraw (PHOTO-PAINT)

.StartDraw .ptX = *long*, .ptY = *long*, .Timer = *long*, .Pressure = *long,* .Tilt = *long,* .Rotate = *long*

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

Parameter	Description
.ptX	Lets you specify the X-coordinate of the point to start drawing. For Brush, Effects and Clone tool, this value is expressed in 1/256 of a pixel, relative to the origin. For other tools, this value is expressed in pixels, relative to the origin.
.ptY	Lets you specify the Y-coordinate of the point to start drawing. For Brush, Effects and Clone tool, this value is expressed in 1/256 of a pixel, relative to the origin. For other tools, this value is expressed in pixels, relative to the origin.
.Timer	Used for tools like the airbrush, where the nib size changes with time. Values start at 0 and increment by 1 for every clock cycle.
.Pressure	Used for a Pen Tablet that has pressure sensitivity. Values range from 0 to 9999.
.Tilt	Lets you specify the tilt attribute of the clone brush tool. Values range from 0-90.
.Rotate	Lets you specify the rotation angle of the clone brush tool. Values range from 0-360.

Example

```
.ImageSprayerTool "fruit.cpt", 1, 5, 87, 0, 0
    .ImageSprayerSettings 1, 11, 19, 0, 0, 1, 5, 1
    .RandomSeed 552585248
    .StartDraw 11904, 8320, 0, 0
    .ContinueDraw 14343, 9096, 0, 0
    ...
    .ContinueDraw 61761, 18445, 0, 0
    .EndDraw
```

This example draws a brush stroke containing a variety of fruits.

```
.Eraser 20, 100, 0, 0, 0, 0, 0
    .SetPaperColor 5, 255, 255, 255, 0
    .StartDraw 40320, 8832, 0, 0
    .ContinueDraw 39040, 8832, 0, 0
    ...
    .ContinueDraw 59678, 18166, 0, 0
    .EndDraw
```

This example erases a portion of your image defined by the draw path.

{button ,AL(`OVR1 Drawing commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Tool commands (PHOTO-PAINT)

BrushDabSettings (PHOTO-PAINT)

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

This command sets the attributes for brush dabs.

Parameter	Description
.Dabs	Lets you specify the number of parallel strokes drawn. Valid values range from 1 to 25.
.Spacing	Lets you specify the spacing between dabs as a percentage of brush size. Valid values range from 1 to 999 pixels.
.Spread	Lets you specify maximum random deviation of the dab positions as a percentage of brush size. Valid values range from 0 to 999. Its effect changes depending whether the .Dabs parameter is equal to 1 or greater than 1. If the .Dabs parameter equals 1, each time a dab is drawn its position is varied randomly. If the .Dabs parameter is greater than 1, the initial positions of the strokes are selected randomly, and they maintain their relative positions for the rest of the stroke.
.FadeOut	Lets you specify 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 0 to 100.
.Hue	Lets you specify the random variation in the hue (a particular color) when each dab is drawn, as a percentage. Valid values range from 0 to 100.
.Saturation	Lets you specify the random variation in the saturation (amount of a color) when each dab is drawn, as a percentage. Valid values range from 0 to 100.
.Luminance	Lets you specify the random variation of color brightness when each dab is drawn, as a percentage. Valid values range from 0 to 100.

Example

```
.SetPaintColor 5, 0, 51, 204, 0
.BrushSettings 25, 0, 1, 0, 20, 0, 20, 0, 0, 100, 80
.BrushTextureSettings , 0, 0, 0, 0, 0, -1
.BrushDabSettings 4, 20, 90, 50, 40, 60, 50
.StartDraw 13, 16
    .ContinueDraw 14, 21
    .ContinueDraw 16, 26
     ...
     ...
    .ContinueDraw 46, 24
    .ContinueDraw 42, 20
.EndDraw
```

This example sets the brush dab settings to 4 parallel strokes, with a spacing of 20, a spread of 90, fade out of 50, hue variation 40%, saturation variation 50%, and luminance variation 60%.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

BrushOrbitSettings (PHOTO-PAINT)

.BrushOrbitSettings .Points = *long*, .Radius = *long*, .Rotation = *long*, .GrowRate = *long*, .GrowScale = *long*, .Center = *boolean*, .Clockwise = *boolean*, .ColorHue = *long*, .ColorSaturation = *long*, .ColorLightness = *long*, .Hue = *long*, .Saturation = *long*, .Lightness = *long*

This command...

Parameter	Description
.Points	*long*
.Radius	*long*
.Rotation	*long*
.GrowRate	*long*
.GrowScale	*long*
.Center	*boolean*
.Clockwise	*boolean*
.ColorHue	*long*
.ColorSaturation	*long*
.ColorLightness	*long*
.Hue	*long*
.Saturation	*long*
.Lightness	*long*

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

BrushTextureSettings (PHOTO-PAINT)

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

This command sets the attributes for the brush texture.

Parameter	Description
.TextureFile	Lets you specify the name of the Texture file and its path (optional). If the default texture is used, TextureFile does not need to be set.
.BrushTexture	Lets you specify the amount of texture of the brush. Increase the value to make the brush stroke more coarse. Valid values range from 0 to 100.
.EdgeTexture	Lets you specify the amount of texture of the brush. Valid values range from 0 to 100.
.Bleed	Lets you specify 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 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	Lets you specify the amount of brush stroke smoothing, in pixels. Smoothing helps to create a more flowing and fluid paint stroke by smoothing out the sharper 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.
.Cumulative	Set to TRUE (-1) to make the effects of brush strokes cumulative. Set to FALSE (0) if you want each brush stroke to "max out" after a certain point. For example, if you are applying a tint to an area and want it to appear uniform.
.Range	Lets you specify the range of effect for the Dodge/Burn tool, where 0 = Highlights 1 = Midtones 2 = Shadows
.MergedSource	Set to TRUE (-1) to use all objects. Set to FALSE (0) to use the active objects only.

Example

```
.SetPaintColor 5, 0, 204, 0, 0
.BrushSettings 25, 4, 0, 0, 20, 2, 2, 6, 10, 95, 30
.BrushTextureSettings , 3, 0, 0, 0, 24, -1
.BrushDabSettings 1, 11, 9, 0, 0, 0, 0
.NibSettings "PNTBRBR.MSK", 19
.StartDraw 89, 26
    .ContinueDraw 89, 27
    .ContinueDraw 90, 28
     ...
     ...
    .ContinueDraw 93, 29
    .ContinueDraw 94, 30
.EndDraw
```

This example sets the brush texture to the default texture with coarseness set to 3, 24 percent smoothing and anti-aliasing enabled.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

BrushTool (PHOTO-PAINT)

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

This command creates a brush stroke with one of the predefined brushes. A BrushTool command block must end with a PathStroke command.

Parameter	Description
.BrushID	Lets you specify the brush to use by index number.
.TypeID	Lets you specify the type of brush stroke by index number.
.MergeMode	Lets you specify 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	Lets you specify the rate at which the effect or paint is applied to the image, ranging from 1 to 100. A higher value results in a more pronounced effect or heavier application of paint.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency level of the nib. Valid values range from 0 to 99%.
.Rotate	Lets you specify the angle at which the nib is rotated. Valid values range from 0 to 360 degrees.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 99.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.BrushTool 0, 0, 0, 20, 0, 5, 0, 0, 100, 0
    .BrushTextureSettings "", 0, 0, 0, 0, 0, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 186, 159, 106, 0
    .PathStroke
```

This example draws a brush stroke.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

CloneTool (PHOTO-PAINT)

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

This command copies a part of your image to another part, following a path defined by StartCloneDraw and ContinueDraw commands. A CloneTool command block must end with an EndDraw command.

Parameter	Description
.BrushID	Lets you specify the brush to use by index number.
.TypeID	Lets you specify the type of brush stroke by index number.
.MergeMode	Lets you specify 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	Lets you specify the rate at which the effect or paint is applied to the image, ranging from 1 to 100. A higher value results in a more pronounced effect or heavier application of paint.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency of the nib. Valid values range from 0 to 100%.
.Rotate	Lets you specify the rotation of the nib. Valid values range from 0 to 360 degrees.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 100%.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.CloneTool 0, 0, 0, 20, 1, 33, 0, 0, 100, 80
    .BrushTextureSettings "", 0, 0, 0, 0, 10, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 186, 159, 106, 0
    .RandomSeed 721390233
    .StartCloneDraw 28, 250, 38016, 23168, 0, 0
    .ContinueDraw 39046, 25228, 0, 0
    ...
    .ContinueDraw 55430, 26365, 0, 0
    .EndDraw
```

This example clones one part of your image to another part.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ColorReplace (PHOTO-PAINT)

.ColorReplace

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

Parameter	Description
.ToleranceMode	Lets you specify the tolerance mode: 0 = Normal 1 = HSB
.Normal	If ToleranceMode is set to 0, specifies the tolerance as a percentage (0-100). If ToleranceMode is set to 1 (HSB mode).
.Hue	Lets you specify the hue tolerance as a percentage (0-100). 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. This parameter is used only if ToleranceMode is set to 1.
.Saturation	Lets you specify the saturation tolerance as a percentage (0-100). 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. This parameter is used only if ToleranceMode is set to 1.
.Brightness	Lets you specify the brightness tolerance as a percentage (0-100). In the HSB color model, the component that determines the amount of black in a color. This parameter is used only if ToleranceMode is set to 1.

Example

```
.ColorReplace 0, 10, 10, 10, 10
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ColorReplacerTool (PHOTO-PAINT)

.ColorReplacerTool .Width = *long*, .Flatten = *long*, .Rotate = *long*, .NibShape = *long*, .Transparency = *long*, .SoftEdge = *long*

This command sets the attributes of the Color Replacer tool.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 100%.
.Rotate	Lets you specify the rotation of the nib. Valid values range from 0 to 360 degrees.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Transparency	Lets you specify the transparency of the nib. Valid values range from 0 to 100%.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.ColorReplacerTool 20, 100, 0, 0, 0, 80, 0
    .SetPaintColor 5, 255, 255, 0, 0
    .SetPaperColor 5, 255, 153, 204, 0
    .ToleranceSettings 0, 10, 0, 0, 0
    .StartDraw 50773, 36295, 0, 0
    .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndObjectColorTransparencyTool (PHOTO-PAINT)

.EndObjectColorTransparencyTool

This command completes the .ObjectColorTransparencyTool operation.

Example

```
.ObjectColorTransparencyTool 0, TRUE, 27
    .ColorMaskColor 0, 5, 62, 155, 0, 0, 14, 30, 50, 10
    .ColorMaskColor 1, 5, 62, 155, 0, 0, 14, 30, 50, 10
    .ColorMaskColor 2, 5, 112, 102, 28, 0, 14, 30, 50, 10
    .EndObjectColorTransparencyTool
```

This example makes the selected colors transparent in the currently selected object using a clip mask.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectPlugin (PHOTO-PAINT)

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

This command activates the specified plug-in effect.

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

Note

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

Example

```
.EffectPlugin "3D Effects", "Emboss ...", 1, 56,
"db00000001000000030000000000000001000000010000003200000032000000 7f9099050000000000000000000000
0000000000000000000"
```

This example activates the Emboss effect.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EllipseTool (PHOTO-PAINT)

.EllipseTool .Width = *long*, .Transparency = *long*, .MergeMode = *long*, .AntiAlias = *boolean*, .RenderObject = *boolean*, .Fill = *boolean*

This command sets the attributes of the Ellipse tool.

Parameter	Description
.Width	Lets you specify the width of the shape's outline. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency of the shape's outline. Valid values range from 0 to 100%.
.MergeMode	Lets you specify 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
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.RenderObject	Set to TRUE (-1) to convert the shape to an object.
.Fill	Set to TRUE (-1) to fill the shape with the current fill. Set to FALSE (0) to leave shape unfilled.

Example

```
.EllipseTool 0, 0, 0, TRUE, TRUE, TRUE
    .FillBitmap "d:\Fill.cpt", 254000, 254000, 0, 0, FALSE, 0, FALSE, FALSE, TRUE
    .Ellipse 51, 199, 232, 356
```

This example applies the specified bitmap fill inside an ellipse.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillBitmap (PHOTO-PAINT)

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

This command specifies bitmap fill settings to be used by a following Fill, Rectangle, Ellipse or Polygon command.

Parameter	Description
.BitmapName	Lets you specify the name of the bitmap file.
.Width	Lets you specify the width of the bitmap in pixels.
.Height	Lets you specify the height of the bitmap in pixels.
.XOffset	Lets you specify the amount of offset applied to the first tile along the x-axis. Valid values range from 0 to 100.
.YOffset	Lets you specify 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	Lets you specify 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 disable scaling.
.OriginalSize	Set to TRUE (-1) to maintain the original size. Set to FALSE (0) to alter the original size.
SkewAngle	Lets you specify the skew angle. This is equivalent to dragging the skew handles onscreen.
RotateAngle	Lets you specify the angle of rotation for the bitmap fill. This is equivalent to dragging the rotation handles onscreen.

Example

```
.FillTool 0, 0
    .ToleranceSettings 0, 6, 10, 10, 10
    .FillBitmap "d:\corel\graphics8\custom\tiles\breadm.cpt", 338666, 338666, 0, 0, FALSE, 0, FALSE, FALSE, TRUE, 24, 36
    .Fill 82, 99, TRUE
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillFountainApply (PHOTO-PAINT)

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

This command is used with the FillFountainColor command to specify fountain-fill settings in an EditFill command block.

Parameter	Description
.Type	Lets you specify the type of Fountain Fill to apply: 0 = Linear 1 = Radial 2 = Conical 3 = Square
.Colors	Lets you specify the number of colors used for the fill. This number will correspond to the number of calls to the FillFountainColor command.
.Steps	Lets you specify 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	Lets you specify 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	Lets you specify the amount of padding to apply to the fill Ignored for type 2. Valid values range from 0 to 45 percent.
.HorizontalOffset	Lets you specify 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	Lets you specify 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	Lets you specify the location in the fill of the color midway between the two endpoint colors. It is specified as a percentage (1-to-99%). It only applies to 2-color fountain fills.

Example

```
.EditFill 0, 0, 100, 3, 3, 294, 240, 79, 164, 114, 61
    .FillFountainColor 5, 2, 74, 123, 0, 0, 0
    .FillFountainColor 5, 255, 255, 255, 0, 50, 1
    .FillFountainColor 5, 63, 125, 122, 0, 100, 2
    .FillFountainApply 3, 3, 256, 135, 0, 0, 0, 50
    .EndEditFill
```

This example creates an Elliptical Fountain Fill with a starting transparency of 0 and an ending transparency of 100.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillFountainColor (PHOTO-PAINT)

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

This command sets fountain fill colors for an EditFill command block.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.Position	Lets you specify 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	Lets you specify the position of the color in the palette. Valid values range from 0 to 255.

Example

```
.EditFill 0, 0, 100, 3, 3, 294, 240, 79, 164, 114, 61
    .FillFountainColor 5, 2, 74, 123, 0, 0, 0
    .FillFountainColor 5, 255, 255, 255, 0, 50, 1
    .FillFountainColor 5, 63, 125, 122, 0, 100, 2
    .FillFountainApply 3, 3, 256, 135, 0, 0, 0, 50
    .EndEditFill
```

This example creates an Elliptical Fountain Fill with a starting transparency of 0 and an ending transparency of 100.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillSolid (PHOTO-PAINT)

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

This command specifies a solid fill color to be used by an EditFill or object tool command block.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.RectangleTool 0, 0, 0, 0, TRUE, TRUE, TRUE
    .FillSolid 5, 255, 255, 255, 0
    .Rectangle 299, 115, 510, 280
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillTexture (PHOTO-PAINT)

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

This command specifies a texture-fill preset to be used by an EditFill or object tool command block.

Parameter	Description
.LibraryName	Lets you specify the name of the Texture Library.
.TextureName	Lets you specify the name of the texture.
.StyleName	Lets you specify the name of the style.

Example

```
.PolygonTool 0, 0, 0, 0, TRUE, FALSE, TRUE
        .FillTexture "", "Putty 2C", "Cdr5:Putty 2C"
        .StartDraw 88888, 79246, 0, 0
        .ContinueDraw 66133, 102855, 0, 0
        .ContinueDraw 124728, 109966, 0, 0
        .ContinueDraw 125582, 82090, 0, 0
        .ContinueDraw 101688, 81521, 0, 0
        .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillTextureSettings (PHOTO-PAINT)

.FillTextureSettings .Index = *long*, .Value1 = *long*, .Value2 = *long*, Value3 = *long*, Value4 = *long*

This command is used in conjunction with the Texture Fill commands. The parameters set up an internal structure, therefore the command is provided only for advanced users.

Parameter	Description
.Index	Lets you specify the index of the control for the desired texture fill.
.Value1	Lets you specify the numbox value, or R value, depending on the defined texture fill.
.Value2	Lets you specify the G value (if one is required for the defined texture fill).
.Value3	Lets you specify the B value (if one is required for the defined texture fill).
.Value4	This value is usually 0.

Note

If this command is not included with a texture fill, the you get the default Fill for that Texture.

Example

```
.FillTool 0, 0
    .ToleranceSettings 0, 100, 10, 10, 10
    .FillTexture "", "Putty 2C", "Cdr5:Putty 2C"
    .FillTextureSettings 0, 28314, 0, 0, 0
    .FillTextureSettings 1, 25, 0, 0, 0
    .FillTextureSettings 2, 75, 0, 0, 0
    .FillTextureSettings 3, 51, 0, 0, 0
    .FillTextureSettings 4, 87, 0, 0, 0
    .FillTextureSettings 5, 52, 0, 0, 0
    .FillTextureSettings 6, 98, 0, 0, 0
    .FillTextureSettings 7, 12, 0, 0, 0
    .FillTextureSettings 8, 73, 40, 68, 0
    .FillTextureSettings 9, 78, 100, 74, 0
    .Fill 338, 177, TRUE
```

This example has ten texture parameters.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

FillTool (PHOTO-PAINT)

.FillTool .Transparency = *long*, .MergeMode = *long*

This command creates a fill in the current image.

Parameter	Description
.Transparency	Lets you specify the transparency of the fill. Valid values range from 0 to 100%.
.MergeMode	Lets you specify 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

Example

```
FillTool 0, 0
    .ToleranceSettings 0, 100, 10, 10, 10
    .FillSolid 5, 0, 0, 255, 0
    .Fill 73, 53, TRUE
```

This example applies a solid blue fill beginning at the point (73, 53) with no anti-aliasing. These coordinates are expressed in pixels.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

GradientTool (PHOTO-PAINT)

.GradientTool .Style = *long*, .MergeMode = *long*, .Transparency = *long,* .Handles = *long*

This command draws a gradient in the current image. A GradientTool command block must end with a Gradient command.

Parameter	Description
.Style	Lets you specify the type of transparency pattern: 0 = Paint to Paper 1 = Paint to Transparency 2 = Transparent to Paint 3 = Clockwise Spectrum 4 = Counterclockwise Spectrum
.MergeMode	Lets you specify 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
.Transparency	Sets the transparency level of the fill. Valid values range from 0 to 100%.
.Handles	Lets you specify the number of handles used to define the gradient.

Example

```
.GradientTool 5, 0, 0, 4
    .SetPaintColor 5, 102, 255, 0, 0
    .SetPaperColor 8, 0, 0, 0, 0
    .GradientPoint 0, 23, 169, 5, 255, 255, 255, 0, 255
    .GradientPoint 1, 65, 133, 5, 153, 255, 0, 0, 255
    .GradientPoint 2, 124, 82, 5, 255, 51, 0, 0, 255
    .GradientPoint 3, 173, 39, 8, 0, 0, 0, 0, 255
    .Gradient 2, 50
```

This example applies a gradient fill to the active objects.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

LineTool (PHOTO-PAINT)

.LineTool .Width = *long*, .Transparency = *long*, .Joints = *long*, .MergeMode = *long*, .AntiAlias = *boolean*, .RenderObject = *boolean*

This command draws a line in your drawing. A LineTool command block must contain StartDraw and ContinueDraw commands, and end with an EndDraw command.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency level of the nib. Valid values range from 0 to 99%.
.Joints	Lets you specify the type of corner joint: 0 = Butt 1 = Filled 2 = Round 3 = Point
.MergeMode	Lets you specify 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
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.RenderObject	Set to TRUE (-1) to convert the line to an object.

Example

```
.LineTool 20, 0, 0, 0, TRUE, FALSE
    .SetPaintColor 5, 255, 255, 0, 0
    .StartDraw 63004, 87779, 0, 0
    .ContinueDraw 127004, 94037, 0, 0
    .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

NibSettings (PHOTO-PAINT)

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

This command is used to select custom nibs.

Parameter	Description
.FileName	Lets you specify the name of the Nib file.
.NibIndex	Identifies the Nib to load. Refer to the Nibs roll-up for more details.

Example

```
.NibSettings "PNTBRUSH.MSK", -1
```

This example selects a custom nib from the specified file.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

PolygonTool (PHOTO-PAINT)

.PolygonTool .Width = *long*, .Transparency = *long*, .Joints = *long*, .MergeMode = *long*, .AntiAlias = *boolean*, .RenderObject = *boolean*, .Fill = *boolean*

This command sets the attributes of the Polygon tool.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency level of the nib. Valid values range from 0 to 99%.
.Joints	Lets you specify the type of corner joint: 0 = Butt 1 = Filled 2 = Round 3 = Point
.MergeMode	Lets you specify 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
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.RenderObject	Set to TRUE (-1) to convert the polygon to an object.
.Fill	Set to TRUE (-1) to fill the shape with the current fill. Set to FALSE (0) to leave shape unfilled.

Example

```
.PolygonTool 0, 0, 0, 0, TRUE, FALSE, TRUE
        .FillTexture "", "Putty 2C", "Cdr5:Putty 2C"
        .StartDraw 88888, 79246, 0, 0
        .ContinueDraw 66133, 102855, 0, 0
        .ContinueDraw 124728, 109966, 0, 0
        .ContinueDraw 125582, 82090, 0, 0
        .ContinueDraw 101688, 81521, 0, 0
        .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

PressureSettings (PHOTO-PAINT)

.PressureSettings .Size = *long*, .Transparency = *long*, .Softness = *long*, .Hue = *long*, .Saturation = *long*, .Brightness = *long*, .Texture = *long*, .Bleed = *long*, .Resaturate = *long*, .Mask = *long,* .Elongation = *long*

This command adjusts the behavior of a drawing tool depending on the pressure applied to a pen tablet.

Parameter	Description
.Size	Lets you specify the ratio of nib size to pen pressure in pixels.
.Transparency	Lets you specify the ratio of transparency to pen pressure. Valid values range from 0 to 100%.
.Softness	Lets you specify the ratio of softness to pen pressure. Valid values range from 0 to 100%.
.Hue	Lets you specify the ratio of hue tolerance to pen pressure. Valid values range from -360 to 360 degrees.
.Saturation	Lets you specify the ratio of saturation tolerance to pen pressure. Valid values range from 0 to 100%.
.Brightness	Lets you specify the ratio of brightness tolerance to pen pressure. Valid values range from 0 to 100%.
.Texture	Lets you specify the ratio of texture density to pen pressure when painting with a texture fill. Valid values range from 0 to 100%.
.Bleed	Lets you specify the ratio of bleed to pen pressure. Valid values range from 0 to 100%.
.Resaturate	Lets you specify the sustain color value, such as found in the tool settings dialog. This is the rate at which your pen runs out of ink near the end of a stroke.
.Mask	Lets you specify which of the above settings to apply. Add any two index numbers together to apply both settings: 0 = None 1 = Size 2 = Transparency 4 = Softness 8 = Hue 16 = Saturation 32 = Brightness 64 = Texture 128 = Bleed 256 = Resaturate
.Elongation	Lets you specify the tilt and rotation attributes for the pressure-sensitive pen. Values range from 0-999.

Example

```
BrushTool 0, 1, 0, 20, 0, 20, 0, 0, 100, 80
    .BrushTextureSettings "", 0, 0, 0, 0, 0, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 0, 0, 0, 0
    .PressureSettings 94, -77, 27, 36, 199, 153, 48, 83, 100, 511
    .RandomSeed 450724147
    .StartDraw 11904, 22400, 0, 180
    .ContinueDraw 13104, 22844, 0, 336
    .ContinueDraw 14424, 23333, 0, 508
    ......
    .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

RandomSeed (PHOTO-PAINT)

.RandomSeed .Seed = *long*

This command sets the random seed for the tools.

Parameter	Description
.Seed	Lets you specify the value of the random seed. Values range from 0 to 2, 147, 483. 647.

Example

```
.BrushTool 0, 0, 0, 20, 0, 19, 50, 0, 100, 77
        .BrushTextureSettings "", 0, 0, 0, 0, 12, TRUE, FALSE, 0, FALSE
        .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
        .BrushOrbitSettings 0, 25, 10, 15, 0, TRUE, FALSE, 100, 200, 200, 0, 0, 0
        .SymmetrySettings 251657, 275250, 770, 9
        .SetPaintColor 5, 102, 255, 0, 0
        .RandomSeed 523388952
        .StartDraw 18112, 16960, 0, 0
        .ContinueDraw 16880, 17305, 0, 0
        ......
        .ContinueDraw 22257, 23507, 0, 0
        .ContinueDraw 23444, 23030, 0, 0
        .ContinueDraw 24722, 22978, 0, 0
        .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

RectangleTool (PHOTO-PAINT)

.RectangleTool .Width = *long*, .Transparency = *long*, .Roundness = *long*, .MergeMode = *long*, .AntiAlias = *boolean*, .RenderObject = *boolean*, .Fill = *boolean*

This command draws a rectangle in the current image.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency level of the nib. Valid values range from 0 to 99%.
.Roundness	Lets you specify the curvature of the rectangle's corners. Valid values range from 0 to 100.
.MergeMode	Lets you specify 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
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.RenderObject	Set to TRUE (-1) to convert the rectangle to an object.
.Fill	Set to TRUE (-1) to fill the shape with the current fill. Set to FALSE (0) to leave shape unfilled.

Example

```
.RectangleTool 0, 0, 0, 0, -1, -1, -1
    .FillSolid 3, 0, 0, 255, 0
    .Rectangle 20, 20, 200, 200
```

This example creates a solid blue rectangle.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

RepeatSettings (PHOTO-PAINT)

.RepeatSettings .Repeat = *long,* .Rotate = *long*, .RotateVar = *long*, .ColorFromImage = *long*, .Hue = *long*, .Lightness = *long*, .Saturation = *long*, .AccumulateAngle = *long*, .TangentToPath = *long*, .Scale = *long*, .ScaleVar = *long*, .StrokePath = *long*, .FromMask = *boolean,* .Left = *long*, .Top = *long*, .Right = *long*, .Bottom = *long*,

This command repeats the currently loaded path as a brush stroke.

Parameter	Description
.Repeat	Lets you specify the number of times to repeat the brush stroke.
.Rotate	Lets you specify the rotation of the brush stroke. Valid values range from 0 to 360 degrees.
.RotateVar	Lets you specify the variation of each successive rotation. Valid values range from 0 to 360 degrees.
.ColorFromImage	Lets you specify the color to use for the brush stroke: 0 = Uses the current paint color 1 = Takes the color that appears in the image at the starting point of the brush stroke
.Hue	Lets you specify the variation in the Hue color channel of each brush stroke. Valid values range from 0 to 100%.
.Lightness	Lets you specify the variation in the Lightness color channel of each brush stroke. Valid values range from 0 to 100%.
.Saturation	Lets you specify the variation in the Saturation color channel of each brush stroke. Valid values range from 0 to 100%.
.AccumulateAngle	Lets you specify the rotation setting: 0 = Rotates each successive brush stroke by the .Rotate value 1 = Increments the rotation by the .Rotate value each time
.TangentToPath	Lets you specify the direction of the brush stroke: 0 = Strokes with the path 1 = Strokes perpendicular to the path
.Scale	Lets you specify the size of the brush stroke relative to the original path.
.ScaleVar	Lets you specify how much each brush stroke can vary in size.
.StrokePath	Lets you specify whether to use the stroke path: 0 = Applies the brush stroke within the current bounding box only 1 = Strokes along the current path
.FromMask	Set to TRUE (-1) to apply paint strokes to the edge of the current mask. Otherwise, set to FALSE (0).
.Left	Lets you specify the horizontal coordinate of the top-left corner of the bounding box.
.Top	Lets you specify the vertical coordinate of the top-left corner of the bounding box.
.Right	Lets you specify the horizontal coordinate of the bottom-right corner of the bounding box.
.Bottom	Lets you specify the vertical coordinate of the bottom-right corner of the bounding box.

Example

```
.BrushTool 0, 0, 0, 20, 0, 20, 0, 0, 100, 25
    .BrushTextureSettings "", 0, 0, 0, 0, 0, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 255, 255, 0, 0
    .RandomSeed 1090138539
    .RepeatSettings 1, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 127, 281, 313, 409
    .StartDraw 78080, 71936, 0, 0
    .ContinueDraw 78080, 70656, 0, 0

    .ContinueDraw 77090, 50207, 0, 0
    .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

SetPaintColor (PHOTO-PAINT)

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

This command lets you set the paint color used by the Paint tool.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

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

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

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

SetPaperColor (PHOTO-PAINT)

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

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

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

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

This example sets the paper color to blue.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

SymmetrySettings (PHOTO-PAINT)

.SymmetrySettings .CenterX = *long*, .CenterY = *long*, .Mode = *long*, .NumPoints = *long*

This command sets symmetry properties.

Parameter	Description
.CenterX	Sets the x coordinate of the symmetry center.
.CenterY	Sets the y coordinate of the symmetry center.
.Mode	Lets you specify the mode 0 = None 1 = Radial 2 = Mirror If you want to set vertical symmetry only, add 512 to the mode value. If you want to set horizontal symmetry only, add 256 to the mode value.
.NumPoints	Lets you specify the number of radial points.

Example

```
.BrushTool 0, 0, 0, 20, 0, 19, 50, 0, 100, 77
        .BrushTextureSettings "", 0, 0, 0, 0, 12, TRUE, FALSE, 0, FALSE
        .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
        .BrushOrbitSettings 0, 25, 10, 15, 0, TRUE, FALSE, 100, 200, 200, 0, 0, 0
        .SymmetrySettings 251657, 275250, 770, 9
        .SetPaintColor 5, 102, 255, 0, 0
        .RandomSeed 523388952
        .StartDraw 18112, 16960, 0, 0
        .ContinueDraw 16880, 17305, 0, 0
        ......
        .ContinueDraw 22257, 23507, 0, 0
        .ContinueDraw 23444, 23030, 0, 0
        .ContinueDraw 24722, 22978, 0, 0
        .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

TextSettings (PHOTO-PAINT)

.TextSettings .Bold = *long*, .Italic = *boolean*, .Underline = *boolean*, .Alignment = *long*, .FontName = *string*, .FontSize = *long*, .AntiAlias = *boolean*, .CharSpacing = *long*, .LineSpacing = *long*, .DrawMode = *long*, .RenderMask = *boolean*

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

Parameter	Description
.Bold	Lets you specify the text's weight setting. The minimum value is 0 and the maximum value is 1000. The setting for a normal weight is 400. The setting for a bold weight is 700.
.Italic	Set to TRUE (-1) to change text to italics.
.Underline	Set to TRUE (-1) to underline text.
.Alignment	Lets you specify the text alignment: 0 = Left 1 = Center 2 = Right
.FontName	A string specifying the font name. Installed fonts vary depending on your system.
.FontSize	Lets you specify the font size, in points.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.CharSpacing	Lets you specify the spacing between characters. Valid values range from 0 to 100%.
.LineSpacing	Lets you specify the spacing between lines. Valid values range from 0 to 100%.
.DrawMode	Lets you specify the draw mode 0 = Normal 1 = Add 2 = Subtract 3 = XOR
.RenderMask	Set to TRUE (-1) to render the text as a mask. Set to FALSE (0) to render the text as a normal object.

Example

```
.TextTool 65, 94, "An example"
    .SetPaintColor 5, 186, 159, 106, 0
    .TextSettings 700, FALSE, FALSE, 0, "News701 BT", 44, TRUE, 26, 85, 0, FALSE
```

This example writes the words "An example" into your image.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

TextTool (PHOTO-PAINT)

.TextTool .ptX = *long*, .ptY = *long*, .Text = *string*

This command inserts a text object at the specified point in your graphic.

Parameter	Description
.ptX	Lets you specify the X-coordinate of the upper-left corner for the new text object in pixels, relative to the origin.
.ptY	Lets you specify the Y-coordinate of the upper-left corner for the new text object in pixels, relative to the origin.
.Text	A string specifying the contents of the text object.

Example

```
.TextTool 65, 94, "An example"
    .SetPaintColor 5, 186, 159, 106, 0
    .TextSettings 700, FALSE, FALSE, 0, "News701 BT", 44, TRUE, 26, 85, 0, FALSE
```

This example writes the words "An example" into your image.

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

ToleranceSettings (PHOTO-PAINT)

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

This command specifies the color tolerance to be used by the Magic Wand and Fill tools.

Parameter	Description
.ToleranceMode	Lets you specify the tolerance mode: 0 = Normal 1 = HSB
.Normal	If ToleranceMode is set to 0, specifies the tolerance as a percentage (0-100). If ToleranceMode is set to 1 (HSB mode).
.Hue	Lets you specify the hue tolerance as a percentage (0-100). 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. This parameter is used only if ToleranceMode is set to 1.
.Saturation	Lets you specify the saturation tolerance as a percentage (0-100). 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. This parameter is used only if ToleranceMode is set to 1.
.Brightness	Lets you specify the brightness tolerance as a percentage (0-100). In the HSB color model, the component that determines the amount of black in a color. This parameter is used only if ToleranceMode is set to 1.

Example

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

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

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

TransparencyBrushTool (PHOTO-PAINT)

.TransparencyBrushTool .Width = *long*, .Flatten = *long*, .Rotate = *long*, .NibShape = *long*, .Transparency = *long*, .SoftEdge = *long*, .Opacity = *long*, .UseOriginal = *boolean,* AntiAlias = *boolean,* ApplyToClipMask = *boolean*

This command makes a part of your drawing more transparent, following a brush stroke defined by a series of Draw commands. A TransparencyBrushTool command block must contain a series of StartDraw and ContinueDraw commands, and end with an EndDraw command.

Parameter	Description
.Width	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 100%.
.Rotate	Lets you specify the rotation of the nib. Valid values range from 0 to 360 degrees.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Transparency	Lets you specify the transparency of the nib. Valid values range from 0 to 100%.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.
.Opacity	Lets you specify the maximum opacity level to apply to pixels repeatedly brushed with the Transparency Brush tool.
.UseOriginal	Set to TRUE (-1) to add the transparency value you set to the transparency value of the pixels in the object. Set to FALSE (0) to replace the existing transparency values of the object's pixels with the values set for this tool.
.AntiAlias	Set to TRUE (-1) to apply anti-aliasing. Set to FALSE (0) to disable anti-aliasing.
.ApplyToClipMask	Set to TRUE (-1) to apply the transparency to the object's clip mask. Otherwise, set to FALSE (0).

Example

```
.TransparencyBrushTool 20, 100, 0, 0, 0, 80, 76, FALSE, TRUE, FALSE
    .StartDraw 24640, 19264, 0, 0
    .ContinueDraw 24556, 20541, 0, 0
    .ContinueDraw 24512, 21820, 0, 0
    .ContinueDraw 24512, 23100, 0, 0
    .....
    .ContinueDraw 33652, 22907, 0, 0
    .ContinueDraw 34928, 23005, 0, 0
    .EndDraw
```

{button ,AL(`OVR1 Tool commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Effect commands (PHOTO-PAINT)

AdjustEffectInfo (PHOTO-PAINT)

.AdjustEffectInfo .EffectID = *long*, .Value1 = *long*, .Value2 = *long*, .Value3 = *long*

This command sets the individual effect attributes for an AdjustEffect command. This command must appear in a block starting with one of the AdjustEffect commands.

Parameter	Description
.EffectID	Lets you specify the effect to adjust: None = 0 Dirsmooth = 1 Smooth = 2 Soften = 3 Gaussblur = 4 Motionblur = 5 Gaussnoise = 6 Spikenoise = 7 Uniformnoise = 8 Diffuse = 9 Jagdespeck = 10 Removenoise = 11 Minimum = 12 Median = 13 Maximum = 14 Aunsharp = 15 Dirsharpen = 16 Edgeenhance = 17 Sharpen = 18 Unsharpmask = 19 Motionblurangle = 20
.Value1	Lets you specify the first value of the adjusted effect. These values vary with the effect being adjusted.
.Value2	Lets you specify the second value of the adjusted effect. These values vary with the effect being adjusted.
.Value3	Lets you specify the third value of the adjusted effect. These values vary with the effect being adjusted.

Example

```
.EffectAdjustBlur
    .AdjustEffectInfo 4, 10, 0, 0
    .AdjustEffectInfo 20, 10, 0, 0
    .AdjustEffectInfo 20, 10, 69, 0
    .AdjustEffectInfo 1, 10, 69, 0
    .EndAdjustEffect
```

This example adjusts the Blur effect.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Effect3DRotate (PHOTO-PAINT)

.Effect3DRotate .Horizontal = *long*, .Vertical = *long*, .Face = *long*, .BestFit = *boolean*

This command rotates the image horizontally and vertically according to the horizontal and vertical parameters you set. The image is rotated as if it were one side of a three-dimensional box.

Parameter	Description
.Horizontal	Lets you specify the degree of horizontal rotation. Valid values range from -75 to 75.
.Vertical	Lets you specify the degree of vertical rotation. Valid values range from -75 to 75.
.Face	Lets you specify the face of the rotation cube that faces forward. Valid values range from 0 to 4.
.BestFit	Set to TRUE (-1) if you want to ensure that all parts of your image remain within the Image Window.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

Effect3DStereoNoise (PHOTO-PAINT)

.Effect3DStereoNoise .Depth = *long*, .Dots = *boolean*

This command generates a dithered noise pattern. The result is an image that has the appearance of 3D depth when viewed a certain way.

Parameter	Description
.Depth	Lets you specify the depth of the stereogram image. Valid values range from 1 to 9.
.Dots	Set to TRUE (-1) to display dots to help focus on the stereogram image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectAdaptiveUnsharp (PHOTO-PAINT)

.EffectAdaptiveUnsharp .Percentage = *long*

This command accentuates edge detail by analyzing the pixel value of neighboring pixels.

Parameter	Description
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectAddNoise (PHOTO-PAINT)

.EffectAddNoise .Level = *long*, .Density = *long*, .ColorNoise = *boolean*, .NoiseType = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command creates a granular effect that adds a texture to a flat or overly blended image.

Parameter	Description
.Level	Lets you specify how much noise to add. Valid values range from 0 to 100.
.Density	Lets you specify the density of noise addition. Valid values range from 0 to 100.
.ColorNoise	Set to TRUE (-1) to apply randomly colored noise to the image. Set to FALSE (0) for monochrome noise.
.NoiseType	Lets you specify the type of noise: 0 = Gaussian. Prioritizes colors along a Gaussian curve 1 = Spike. Uses colors that are distributed around a narrow curve 2 = Uniform. Provides an overall granular appearance
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectAdjustBlur (PHOTO-PAINT)

.EffectAdjustBlur

This command blurs the current image using the blur settings specified by AdjustEffectInfo commands. An EffectAdjustBlur command block must end with an EndAdjustEffect command.

Example

```
.EffectAdjustBlur
    .AdjustEffectInfo 4, 10, 0, 0
    .AdjustEffectInfo 20, 10, 0, 0
    .AdjustEffectInfo 20, 10, 69, 0
    .AdjustEffectInfo 1, 10, 69, 0
    .EndAdjustEffect
```

This example adjusts the Blur effect.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectAdjustNoise (PHOTO-PAINT)

.EffectAdjustNoise

This command adds noise to the current image using the settings specified by AdjustEffectInfo commands. An EffectAdjustNoise command block must end with an EndAdjustEffect command.

Example

```
.EffectAdjustNoise
    .AdjustEffectInfo 6, 10, 10, 0
    .AdjustEffectInfo 13, 10, 10, 0
    .AdjustEffectInfo 14, 10, 10, 0
    .AdjustEffectInfo 8, 10, 10, 0
    .AdjustEffectInfo 6, 10, 10, 0
    .AdjustEffectInfo 9, 10, 17, 0
    .EndAdjustEffect
```

This example adds noise to the current image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectAdjustSharpness (PHOTO-PAINT)

.EffectAdjustSharpness

This command sharpens the current image using the settings specified by AdjustEffectInfo commands. An EffectAdjustSharpness command block must end with an EndAdjustEffect command.

Example

```
.EffectAdjustSharpness
    .AdjustEffectInfo 19, 10, 50, 0
    .AdjustEffectInfo 17, 10, 50, 0
    .AdjustEffectInfo 16, 10, 50, 0
    .AdjustEffectInfo 18, 19, 50, 0
    .EndAdjustEffect
```

This example sharpens the current image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectBandPass (PHOTO-PAINT)

.EffectBandPass .InRadius = *long*, .OutRadius = *long*, .InBand = *long*, .MidBand = *long*, .OutBand = *long*

This command adjusts the balance of sharp and smooth areas in an image.

Parameter	Description
.InRadius	Lets you specify the size of the inner band radius. The values range from 1 to 256.
.OutRadius	Lets you specify the size of the outer band radius. The values range from 1 to 256.
.InBand	Lets you specify the weighting of the inner band. To eliminate the sharp or smooth areas within a band, set the weighting to 0. Experiment with different weightings to see which provide the best results. The values range from 0 to 100%.
.MidBand	Lets you specify the weighting of the middle band. The values range from 0 to 100%.
.OutBand	Lets you specify the weighting of the outer band. The values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectBitPlanes (PHOTO-PAINT)

.EffectBitPlanes .Red = *long*, .Green = *long*, .Blue = *long*

This command reduces the image to basic RGB color components and emphasizes tonal changes. For example, certain areas appear as solid blocks because there is little change in tone.

Parameter	Description
.Red	Lets you specify the color sensitivity in the red channel. Valid values range from 0 to 7.
.Green	Lets you specify the color sensitivity in the green channel. Valid values range from 0 to 7.
.Blue	Lets you specify the color sensitivity in the blue channel. Valid values range from 0 to 7.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectCanvas (PHOTO-PAINT)

.EffectCanvas .Filename = *string*, .Transparency = *long*, .Emboss = *long*, .X = *long*, .Y = *long*, .Mode = *long*, .Offset = *long*

This command applies a textured surface over top of an image.

Parameter	Description
.Filename	Lets you specify the name of the file to use as the canvas surface.
.Transparency	Lets you specify the transparency of the canvas effect. Valid values range from 0 to 100%.
.Emboss	Lets you specify the degree of embossing. Embossing gives the canvas a raised, relief effect. Valid values range from 0 to 100%.
.X	Lets you specify the horizontal offset of the canvas map. Valid values range from 0 to 100%.
.Y	Lets you specify the vertical offset of the canvas map. Valid values range from 0 to 100%.
.Mode	Lets you specify the offset mode: 0 = Rows. Offsets rows of tiles 1 = Columns. Offsets columns of tiles 2 = Stretch To Fit
.Offset	Lets you specify the degree of offset. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectDiffuse (PHOTO-PAINT)

.EffectDiffuse .Level = *long*

This command spreads out the pixels of your image to fill in blank spaces and remove noise. Depending on the level you select, the effect can appear smooth, blurry, or produce a soft, double-edged look as if the image were being seen through a photographer's diffusion lens.

Parameter	Description
.Level	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectDirectionalSharpen (PHOTO-PAINT)

.EffectDirectionalSharpen .Percentage = *long*

This command analyzes pixels of similar shades to determine the direction in which to apply the greatest amount of sharpening.

Parameter	Description
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectDirectionalSmooth (PHOTO-PAINT)

.EffectDirectionalSmooth .Percentage = *long*

This command analyzes the value of pixels of similar tonal values to determine the direction in which to apply the greatest amount of smoothing. This subtly smoothes edges and surfaces, giving them anti-aliased edges without distorting the image.

Parameter	Description
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectDisplace (PHOTO-PAINT)

.EffectDisplace .Filename = *string*, .Displacement = *long*, .Edges = *long*, .Horizontal = *long*, .Vertical = *long*

This command alters an image using a displacement map. Corel PHOTO-PAINT includes a number of sample displacement maps you can use; however, you can load any bitmap image as a displacement map. The Displace filter evaluates the color value of pixels in both images, and then shifts the active image according to the values of the displacement map. The result is that values from the displacement map appear as forms, colors, and warp patterns in your image.

Parameter	Description
.Filename	Lets you specify the name of the image file to use as a displacement map.
.Displacement	Lets you specify the scaling mode of the displacement: 0 = Tiles the displacement map over the image 1 = Stretches the displacement map to cover the entire image
.Edges	Lets you specify the method of filling empty areas created by the displacement: 0 = Stretches the edge areas 1 = Wraps the opposite edge around to the empty areas
.Horizontal	Lets you specify the horizontal shift. Valid values range from 0 to 100.
.Vertical	Lets you specify the vertical shift. Valid values range from 0 to 100.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectDustScratch (PHOTO-PAINT)

.EffectDustScratch .Level = *long*, .Radius = *long*

This command reduces image noise by averaging pixel values. This works something like adding water to a dry watercolor painting; adjacent colors bleed into each other. As the name implies, this command is extremely useful for eliminating dust and scratch faults in an image.

Parameter	Description
.Level	Lets you specify how great a change in value must occur to any pixel before the effect is applied. Valid values range from 0 to 255.
.Radius	Lets you specify the range of the effect. Larger values increase the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectEdgeDetect (PHOTO-PAINT)

.EffectEdgeDetect .Sensitivity = *long,* .ColorModel = *long,* .Color1 = *long,* .Color2 = *long,* .Color3 = *long,* .Color4 = *long*

This command finds the edges of elements in your image, then converts them to lines on a background of a single color, allowing you to add a variety of outline effects to your image.

Parameter	Description
.Sensitivity	Lets you specify the intensity of the effect. Valid values range from 1 to 10.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectEdgeDetect 7, 5, 0, 0, 0, 0
```

This example uses black as the color for the edge effect.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectEmboss (PHOTO-PAINT)

.EffectEmboss .Depth = *long*, .Level = *long*, .Direction = *long*, .Color = *long,* .ColorModel = *long,* .Color1 = *long,* .Color2 = *long,* .Color3 = *long,* .Color4 = *long*

This command transforms your image into a relief, making the details appear as ridges and crevices on a flat surface.

Parameter	Description
.Depth	Lets you specify the depth of the ridges and crevices in the relief.
.Level	Lets you specify the amount of background color the relief will contain.
.Direction	Lets you specify the angle at which the light hits the relief. Valid values range from 0 to 360.
.Color	Lets you specify the emboss color: 0 = The original color 1 = Gray 2 = Black 3 = The current paper color
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectEmboss 1, 500, 318, 1, 5, 255, 255, 255, 0
```

This example uses gray color options with the Emboss effect.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectFindEdges (PHOTO-PAINT)

.EffectFindEdges .Level = *long*, .EdgeType = *long*

This command detects the outlines of forms in your image and converts them to soft or solid lines.

Parameter	Description
.Level	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.
.EdgeType	Lets you specify the type of edge: 0 = Soft, blurry outline 1 = Sharp, crisp outline

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectGaussianBlur (PHOTO-PAINT)

.EffectGaussianBlur .Radius = *double*

This command produces a hazy effect, blurring the image according to a Gaussian distribution, which spreads the pixel information outward using bell-shaped curves.

Parameter	Description
.Radius	Lets you specify the intensity of the effect. Valid values range from 1 to 50.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectGlass (PHOTO-PAINT)

.EffectGlass .Angle = *long*, .BevelWidth = *long*, .Brightness = *long*, .Direction = *long*, .Dropoff = *long*, .Opacity = *long*, .Refraction = *long*, .Sharpness = *long*, .Smooth = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command applies the Glass effect to an image. You must create a mask on the image before applying this effect.

Parameter	Description
.Angle	Lets you specify the angle at which the light bounces off the bevel.
.BevelWidth	Lets you specify the width of the bevel.
.Brightness	Lets you specify the intensity of the light source.
.Direction	Lets you specify the angle at which the light hits the bevel.
.Dropoff	Lets you specify the current edge style of the bevel. Gaussian Flat Mesa
.Opacity	Lets you specify the opacity of the sheet of glass.
.Refraction	Lets you specify the amount of refraction.
.Sharpness	Lets you specify the amount of fading at the edge of the light shaft. A lower value results in a concentrated light source (like a flashlight). A higher value results in a softer, larger light source (like a ceiling light).
.Smooth	Lets you specify the smoothness of the bevel.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectGlass 32, 59, 83, 328, 0, 50, 70, 71, 62, 9, 128, 0, 0, 0
```

This example applies the Glass effect to an image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectGlassBlock (PHOTO-PAINT)

.EffectGlassBlock .Width = *long*, .Height = *long*

This command mimics the effect of viewing an image through a number of thick glass blocks.

Parameter	Description
.Width	Lets you specify the width of the glass blocks. Valid values range from 1 to 100.
.Height	Lets you specify the height of the glass blocks. Valid values range from 1 to 100.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectHalftone (PHOTO-PAINT)

.EffectHalftone .Radius = *long*, .Cyan = *long*, .Magenta = *long*, .Yellow = *long*, .Black = *long*

This command gives your image the appearance of a color halftone

Parameter	Description
.Radius	Lets you specify the dot radius. Valid values range from 2 to 10.
.Cyan	Lets you specify the Cyan channel angle. Valid values range from 0 to 359 degrees.
.Magenta	Lets you specify the Magenta channel angle. Valid values range from 0 to 359 degrees.
.Yellow	Lets you specify the Yellow channel angle. Valid values range from 0 to 359 degrees.
.Black	Lets you specify the Black channel angle. Valid values range from 0 to 359 degrees.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectHighPass (PHOTO-PAINT)

.EffectHighPass .Radius = *long*, .Percentage = *long*

This command removes low-frequency detail and shading. The effect can give an image an ethereal, glowing quality. It emphasizes the highlights and luminous areas of an image. At higher settings, the High Pass effect removes most of the image detail, leaving only the edge details clearly visible. If you only want to emphasize highlights, use lower percentage settings.

Parameter	Description
.Radius	Lets you specify the range of the effect. Larger values increase the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Lets you specify the intensity of the effect. Larger values remove more shadow detail. Valid values range from 0 o 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectImpressionist (PHOTO-PAINT)

.EffectImpressionist .Horizontal = *long*, .Vertical = *long*

This command gives your image the look of an impressionist painting by converting your image to dabs of solid color.

Parameter	Description
.Horizontal	Lets you specify the amount of pixel displacement that occurs along the horizontal axis. Valid values range from 1 to 100.
.Vertical	Lets you specify the amount of pixel displacement that occurs along the horizontal axis. Valid values range from 1 to 100.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectJaggyDespeckle (PHOTO-PAINT)

.EffectJaggyDespeckle .Width = *long*, .Height = *long*

This command scatters colors in an image creating a soft, blurred effect with minimal distortion. It is most effective for removing the jagged edges that can appear in line art or high-contrast images.

Parameter	Description
.Width	Lets you specify the intensity of horizontal color scattering. Valid values range from 1 to 5.
.Height	Lets you specify the intensity of vertical color scattering. Valid values range from 1 to 5.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectLensFlare (PHOTO-PAINT)

.EffectLensFlare .X = *double*, .Y = *double*, .Brightness = *long*, .LensType = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command produces rings of light on your image that simulate the flare that appears on a photograph when the camera is aimed toward a direct bright light.

Parameter	Description
.X	Lets you specify the horizontal coordinate of the flare's center.
.Y	Lets you specify the vertical coordinate of the flare's center.
.Brightness	Lets you specify the intensity of the lens flare. The effect of the brightness setting varies with different lens types.
.LensType	Lets you specify the type of lens, which will affect the appearance of the flare: 0 = 50-300mm zoom 1 = 35mm prime 2 = 105mm prime
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectLensFlare 0.73, 0.77, 118, 0, 5, 255, 255, 255, 0
```

This example creates a white lens flare effect.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectLighting (PHOTO-PAINT)

.EffectLighting .Sources = *long*

This command adds the effect of one or more light sources to your drawing. An EffectLighting command block must contain at least one EffectLightSource command, and end with an EndEffectLighting command.

Parameter	Description
.Sources	Lets you specify the number of light sources to add.

Example

```
.EffectLighting 4
    .EffectLightSource 0, 0.5, 0.5, 0, 75, 180, 90, 40, 180, 100, 0, 100, 0, 0, 0, 255, 0, 0
    .EffectLightSource 1, 0.097166, 1.09717, 1, 60, 15, 25, 95, 45, 80, 0, 100, 0, 0, 0, 255,
255, 51
    .EffectLightSource 2, 0.433, 1.1, 1, 60, 15, 25, 95, 45, 80, 0, 100, 0, 0, 0, 204, 102, 255
    .EffectLightSource 3, 0.766, 1.1, 1, 60, 15, 25, 95, 45, 80, 0, 100, 0, 0, 0, 255, 255, 255
    .EndEffectLighting
```

This example adds four light sources to your drawing.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectLowPass (PHOTO-PAINT)

.EffectLowPass .Radius = *long*, .Percentage = *long*

This command removes sharp edges and detail from an image, leaving smooth gradients and low frequency detail.

Parameter	Description
.Radius	Lets you specify the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Lets you specify the intensity of the effect. Larger values reduce harsh transitions between shadows and highlights. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectMapToObject (PHOTO-PAINT)

.EffectMapToObject .Mode = *long*, .Percentage = *long, .*Quality = *long*

This command creates the illusion that the image has been wrapped around a sphere, or a horizontal or vertical cylinder.

Parameter	Description
.Mode	Lets you specify the mapping mode: 0 = Spherical 1 = Horizontal cylinder 2 = Vertical cylinder
.Percentage	Lets you specify the direction and amount of wrapping. Negative percentage values wrap the image toward the back; positive percentage values wrap the image toward the front. Valid values range from -100 to 100%.
.Quality	Provides a list of preset quality levels that you can use when applying the filter. 0=Draft 1=Better 2=Best

Example

```
.EffectMapToObject 0, 19, 2
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectMaximum (PHOTO-PAINT)

.EffectMaximum .Radius = *long*, .Percentage = *long*

This command removes noise by adjusting pixel values based on the maximum pixel value of neighboring pixels. The command also causes a mild blurring effect if applied in large percentages or more than once. The highest setting will completely obscure your image.

Parameter	Description
.Radius	Lets you specify the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectMedian (PHOTO-PAINT)

.EffectMedian .Radius = *long*, .Percentage = *long*

This command removes noise and detail by averaging the colors of adjacent pixels in the image.

Parameter	Description
.Radius	Lets you specify the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectMeshPoint (PHOTO-PAINT)

.EffectMeshPoint .Row = *long*, .Column = *long*, .X = *double*, .Y = *double*

This command defines the grid points for the EffectMeshWarp command. An EffectMeshWarp command block must contain an EffectMeshPoint command for each point in the warp grid.

Parameter	Description
.Row	Lets you specify the row number of the point you are defining.
.Column	Lets you specify the column number of the point you are defining.
.X	Lets you specify the horizontal coordinate of the point. Valid values range from 0.00 to 1.00.
.Y	Lets you specify the vertical coordinate of the point. Valid values range from 0.00 to 1.00.

Example

```
.EffectMeshWarp 5, 5
    .EffectMeshPoint 0, 0, 0, 0
    .EffectMeshPoint 0, 1, 0.25, 0
    .EffectMeshPoint 0, 2, 0.5, 0
    .EffectMeshPoint 0, 3, 0.75, 0
    .EffectMeshPoint 0, 4, 1, 0
    .EffectMeshPoint 1, 0, 0, 0.25
    .EffectMeshPoint 1, 1, 0.340067, 0.345946
    .EffectMeshPoint 1, 2, 0.52862, 0.297297
    .EffectMeshPoint 1, 3, 0.861953, 0.145946
    .EffectMeshPoint 1, 4, 1, 0.25
    .EffectMeshPoint 2, 0, 0, 0.5
    .EffectMeshPoint 2, 1, 0.25, 0.5
    .EffectMeshPoint 2, 2, 0.579125, 0.556757
    .EffectMeshPoint 2, 3, 0.75, 0.5
    .EffectMeshPoint 2, 4, 1, 0.5
    .EffectMeshPoint 3, 0, 0, 0.75
    .EffectMeshPoint 3, 1, 0.121212, 0.859459
    .EffectMeshPoint 3, 2, 0.5, 0.75
    .EffectMeshPoint 3, 3, 0.818182, 0.881081
    .EffectMeshPoint 3, 4, 1, 0.75
    .EffectMeshPoint 4, 0, 0, 1
    .EffectMeshPoint 4, 1, 0.25, 1
    .EffectMeshPoint 4, 2, 0.5, 1
    .EffectMeshPoint 4, 3, 0.75, 1
    .EffectMeshPoint 4, 4, 1, 1
    .EndEffectMeshWarp
```

This example distorts the current image using a 5 x 5 mesh warp grid.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectMeshWarp (PHOTO-PAINT)

.EffectMeshWarp .Width = *long*, .Height = *long*

This command distorts the current image using a grid of points that are displaced to produce the desired warping effect. An EffectMeshWarp command block must contain an EffectMeshPoint command for each point in the warp grid, and the block must end with an EndEffectMeshWarp command.

Parameter	Description
.Width	Lets you specify the number of columns in the warp grid. Valid values range from 5 to 11.
.Height	Lets you specify the number of rows in the warp grid. Valid values range from 5 to 11.

Example

```
.EffectMeshWarp 5, 5
    .EffectMeshPoint 0, 0, 0, 0
    .EffectMeshPoint 0, 1, 0.25, 0
    .EffectMeshPoint 0, 2, 0.5, 0
    .EffectMeshPoint 0, 3, 0.75, 0
    .EffectMeshPoint 0, 4, 1, 0
    .EffectMeshPoint 1, 0, 0, 0.25
    .EffectMeshPoint 1, 1, 0.340067, 0.345946
    .EffectMeshPoint 1, 2, 0.52862, 0.297297
    .EffectMeshPoint 1, 3, 0.861953, 0.145946
    .EffectMeshPoint 1, 4, 1, 0.25
    .EffectMeshPoint 2, 0, 0, 0.5
    .EffectMeshPoint 2, 1, 0.25, 0.5
    .EffectMeshPoint 2, 2, 0.579125, 0.556757
    .EffectMeshPoint 2, 3, 0.75, 0.5
    .EffectMeshPoint 2, 4, 1, 0.5
    .EffectMeshPoint 3, 0, 0, 0.75
    .EffectMeshPoint 3, 1, 0.121212, 0.859459
    .EffectMeshPoint 3, 2, 0.5, 0.75
    .EffectMeshPoint 3, 3, 0.818182, 0.881081
    .EffectMeshPoint 3, 4, 1, 0.75
    .EffectMeshPoint 4, 0, 0, 1
    .EffectMeshPoint 4, 1, 0.25, 1
    .EffectMeshPoint 4, 2, 0.5, 1
    .EffectMeshPoint 4, 3, 0.75, 1
    .EffectMeshPoint 4, 4, 1, 1
    .EndEffectMeshWarp
```

This example distorts the current image using a 5 x 5 mesh warp grid.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectMinimum (PHOTO-PAINT)

.EffectMinimum .Radius = *long*, .Percentage = *long*

This command darkens an image by adjusting pixel values based on the minimum pixel value of neighboring pixels.

Parameter	Description
.Radius	Lets you specify the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectMotionBlur (PHOTO-PAINT)

.EffectMotionBlur .Speed = *long*, .Direction = *long*, .Method = *long*

This command creates the illusion of movement in an image.

Parameter	Description
.Speed	Lets you specify the degree of image blurring. Valid values range from 1 to 50.
.Direction	Lets you specify the direction of blurring. Valid values range from 0 to 360 degrees.
.Method	Lets you specify the method of off-image sampling: 0 = Ignores pixels outside the image 1 = Uses paper color 2 = Samples nearest edge pixel

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectOffset (PHOTO-PAINT)

.EffectOffset .Horizontal = *long*, .Vertical = *long*, .Shift = *boolean*, .Edges = *long,* .ColorModel = *long,* .Color1 = *long,* .Color2 = *long,* .Color3 = *long,* .Color 4 = *long*

This command corrects image positioning by shifting the image.

Parameter	Description
.Horizontal	Lets you specify the amount of horizontal shifting. Valid values range from -100 to 100.
.Vertical	Lets you specify the amount of vertical shifting. Valid values range from -100 to 100.
.Shift	Set to TRUE (-1) to coordinate the horizontal and vertical shift values with the size of the object. With a vertical shift value of 50, the image will shift along the vertical plane a distance corresponding to exactly one-half the size of the image. Set to FALSE (0) for absolute offsetting.
.Edges	Lets you specify the method used to fill the empty area left behind by the offset: 0 = Wraps the opposite edge around to the empty area 1 = Stretches the edges of the image to fill the empty area 2 = Uses the background paper color
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectOffset 71, -33, TRUE, 2, 5, 0, 0, 255, 0
```

This example applies the Offset effecct with a blue background.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectPageCurl (PHOTO-PAINT)

.EffectPageCurl .Corner = *long*, .Direction = *boolean*, .Width = *long*, .Height = *long*, .Opaque = *boolean*, .CurlColorModel = *long*, .CurlColor1 = *long*, .CurlColor2 = *long*, .CurlColor3 = *long*, .CurlColor4 = *long*, .BackColorModel = *long*, .BackColor1 = *long*, .BackColor2 = *long*, .BackColor3 = *long*, .BackColor4 = *long*

This command gives the impression that a corner of your image has rolled in on itself.

Parameter	Description
.Corner	Lets you specify the corner to curl: 0 = Top-left 1 = Top-right 2 = Bottom-left 3 = Bottom-right
.Direction	Set to TRUE (-1) to have the page curl begin along the top or bottom edge of the image, depending on the .Corner location. Set to FALSE (0) to have the page curl begin along the left or right edge of the image.
.Width	Lets you specify the width of the page curl. Increase the value to extend the page curl along the vertical edge of the image.
.Height	Lets you specify the height of the page curl. Increase the value to extend the page curl along the horizontal edge of the image.
.Opaque	Set to TRUE (-1) to make the curl completely opaque. Set to FALSE (0) to have some of the image show through the curl.
.CurlColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.CurlColor1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.CurlColor2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.CurlColor3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.CurlColor4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.
.BackColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.BackColor1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.BackColor2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.BackColor3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.BackColor4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectPerspective (PHOTO-PAINT)

.EffectPerspective .X1 = *double*, .Y1 = *double*, .X2 = *double*, .Y2 = *double*, .X3 = *double*, .Y3 = *double*, .X4 = *double*, .Y4 = *double*, .BestFit = *boolean*

This command gives your image a sense of three-dimensional depth, as if it were on a flat plane receding into the distance.

Parameter	Description
.X1	The horizontal coordinate of the top-left handle of the distortion rectangle. Valid values range from 0 to 48.
.Y1	The vertical coordinate of the top-left handle of the distortion rectangle. Valid values range from 0 to 48.
.X2	The horizontal coordinate of the top-right handle of the distortion rectangle. Valid values range from 0 to 48.
.Y2	The vertical coordinate of the top-right handle of the distortion rectangle. Valid values range from 0 to 48.
.X3	The horizontal coordinate of the bottom-right handle of the distortion rectangle. Valid values range from 0 to 48.
.Y3	The vertical coordinate of the bottom-right handle of the distortion rectangle. Valid values range from 0 to 48.
.X4	The horizontal coordinate of the bottom-left handle of the distortion rectangle. Valid values range from 0 to 48.
.Y4	The vertical coordinate of the bottom-left handle of the distortion rectangle. Valid values range from 0 to 48.
.BestFit	Set to TRUE (-1) to keep two nodes equidistant at all times.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectPinchPunch (PHOTO-PAINT)

.EffectPinchPunch .Level = *long,* .CenterX = *long,* CenterY = *long*

This command warps your image by either "pinching" the image away from you or "punching" it toward you. Negative values apply a punch effect; positive values apply a pinch effect.

Parameter	Description
.Level	Lets you specify the intensity of the pinch or punch effect. Positive values apply a pinch effect, while negative values apply a punch effect. Valid values range from -100 to 100.
.CenterX	Lets you specify the horizontal position of the pinch or punch's center.
.CenterY	Lets you specify the vertical position of the pinch or punch's center.

Example

```
.EffectPinchPunch -54, 158, 126
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectPixelate (PHOTO-PAINT)

.EffectPixelate .Width = *long*, .Height = *long*, .Opacity = *long*, .Mode = *long*

This command breaks up your image into square, rectangular, or circular cells.

Parameter	Description
.Width	Lets you specify the width of the cells. Valid values range from 1 to 100.
.Height	Lets you specify the height of the cells. Valid values range from 1 to 100.
.Opacity	Lets you specify the opacity of the cells. Valid values range from 1 to 100%.
.Mode	Lets you specify the pixelation mode: 0 = Square 1 = Rectangular 2 = Circular

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectPsychedelic (PHOTO-PAINT)

.EffectPsychedelic .Level = *long*

This command changes the colors in your image to bright, electric colors such as orange, hot pink, cyan, and lime green

Parameter	Description
.Level	Lets you specify the intensity of the effect. Valid values range from 0 to 255.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectPuzzle (PHOTO-PAINT)

.EffectPuzzle .Width = *long*, .Height = *long*, .Offset = *long*, .Fill = *long,* .RandSeed = *long,* .ColorModel = *long,* .Color1 = *long,* .Color2 = *long,* .Color 3 = *long,* .Color4 = *long*

This command breaks down the image into puzzle-like pieces or blocks, resembling a jigsaw puzzle.

Parameter	Description
.Width	Lets you specify the width of the puzzle blocks. Valid values range from 1 to 100.
.Height	Lets you specify the height of the puzzle blocks. Valid values range from 1 to 100.
.Offset	Lets you specify the amount of shifting that occurs. Valid values range from 0 to 200%.
.Fill	Lets you specify the method used to fill the empty area behind the puzzle pieces: 0 = Black fill 1 = White fill 2 = Fill using the current paint color 3 = Use the original image 4 = Use a negative of the original image
.RandSeed	Provides a seed for a random number. You can type any value in this parameter.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectPuzzle 10, 10, 28, 1, 51665343, 5, 0, 0, 0, 0
```

This example applies the Puzzle effect with white background options.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectRadialBlur (PHOTO-PAINT)

.EffectRadialBlur .Mode = *long*, .Radius = *long*, .CenterX = *long*, .CenterY = *long*

This command creates a blurring effect that radiates outward from a central point.

Parameter	Description
.Mode	Lets you specify the radial mode: 0 = Spin 1 = Zoom
.Radius	Lets you specify the radius of the blur. Valid values range from 0 to 100.
.CenterX	Lets you specify the horizontal position of the blur's center.
.CenterY	Lets you specify the vertical position of the blur's center.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectRemoveMoire (PHOTO-PAINT)

.EffectRemoveMoire .Amount = *long*, .Quality = *long*, .FinalResolution = *long*, .OriginalResolution = *long*

This command applies the Remove Moire effect to an image.

Parameter	Description
.Amount	Lets you specify the intensity of the effect. Valid values range from 0 to 10.
.Quality	Lets you specify the level of the effect. Better = 0 Faster = 1
.FinalResolution	Lets you specify the resolution of the image after the filter is applied
.OriginalResolution	Lets you specify the resolution of the image before the filter is applied.

Example

```
.EffectRemoveMoire 6, 1, 75, 75
```

This example applies the Remove Moire effect to an image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectRemoveNoise (PHOTO-PAINT)

.EffectRemoveNoise .Threshold = *long*, .Auto = *boolean*

This command softens the image and reduces the speckled effect that can occur during the scanning or video capturing process. The command compares each pixel to surrounding pixels, and calculates an average. Each pixel whose brightness value exceeds that of the threshold you set are removed.

Parameter	Description
.Threshold	Lets you specify how great a change in value must occur to any pixel before the effect is applied.
.Auto	Set to TRUE (-1) to have Corel PHOTO-PAINT automatically calculate the noise reduction level required to improve image quality. Set to FALSE (0) to use the threshold value above.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectRipple (PHOTO-PAINT)

.EffectRipple .Period = *long*, .Amplitude = *long*, .Angle = *long*, .Distort = *boolean,* .Mode = *long*

This command creates vertical or horizontal rippled waves throughout the image.

Parameter	Description
.Period	Lets you specify the distance between each wave cycle. Valid values range from 1 to 100.
.Amplitude	Lets you specify the amount of displacement created by each wave. Valid values range from 1 to 100.
.Angle	Lets you specify the direction of the ripple effect. Valid values range from 0 to 180 degrees.
.Distort	Set to TRUE (-1) to apply distortion to the ripple.
.Mode	Lets you specify the Ripple mode to use for the effect 0=Single Wave 1=Dual Wave 1:1 2=Dual Wave 2:1

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectSharpen (PHOTO-PAINT)

.EffectSharpen .EdgeLevel = *long*, .Background = *long*, .Intensity = *boolean*

This command accentuates the edges in the image by finding the edges and increasing the contrast between adjacent pixels.

Parameter	Description
.EdgeLevel	Lets you specify the amount of edge sharpening. Valid values range from 0 to 100%.
.Background	Lets you specify how great a change in value must occur to any pixel before the effect is applied.
.Intensity	0 sharpens all values 1 sharpens only intensity

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectShear (PHOTO-PAINT)

.EffectShear .Scale = *long*, .Border = *long*, .Orientation = *long,* ColorModel = *long,* Color1 = *long,* Color2 = *long,* Color3 = *long,* Color4 = *long*

This command distorts the current image in a manner defined by the table values set by the EffectShearTable command. An EffectShear command block must contain an EffectShearTable command for each of the 1024 points in the shear table, and must end with an EndEffectShear command.

Parameter	Description
.Scale	Lets you specify the degree to which the image conforms to the curve. Set the value at 100% to have the image conform completely to the curve. Valid values range from 0 to 100%.
.Border	Lets you specify the method to use when filling the empty space left by the shear command: 0 = Wraps the opposite end of the image around to the empty space 1 = Stretches the image to fill the space 2 = Fills empty areas with the current paint color
.Orientation	Lets you specify the direction of the shear curve: 0 = Horizontal 1 = Vertical
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectShear 50, 2, 1, 5, 255, 51, 0, 0
        .EffectShearTable 0, 512
        .EffectShearTable 1, 517
        ....
        .EffectShearTable 1023, 512
        .EndEffectShear
```

This example applies a shear effect to the active image and specifies red for the undefined areas.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectShearTable (PHOTO-PAINT)

.EffectShearTable .Number = *long*, .Value = *long*

This command sets the shear table values for the EffectShear command. An EffectShear command block must contain an EffectShearTable command for each of the 1024 points in the shear table.

Parameter	Description
.Number	Lets you specify the index number of the point on the shear curve. Valid values range from 0 to 1023.
.Value	Lets you specify the displacement of the shear point. Valid values range from 0 to 1023.

Example

```
.EffectShear 50, 0, 1
    .EffectShearTable 0, 264
    .EffectShearTable 1, 265
    .EffectShearTable 2, 267
    .EffectShearTable 3, 269
    ...
    .EffectShearTable 1021, 779
    .EffectShearTable 1022, 780
    .EffectShearTable 1023, 781
    .EndEffectShear
```

This example applies a shear effect to the active image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectSmokedGlass (PHOTO-PAINT)

.EffectSmokedGlass .Tint = *long*, .Percent = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command applies a transparent, colored tint over the image.

Parameter	Description
.Tint	Lets you specify the opacity of the tint. Valid values range from 0 to 100%.
.Percent	Lets you specify the amount of blurring to use to create the glass effect. Valid values range from 0 to 100%.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectSmooth (PHOTO-PAINT)

.EffectSmooth .Percentage = *long*

This command tones down differences in adjacent pixels resulting in only a slight loss of detail, while smoothing the overall image or selected area.

Parameter	Description
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectSoften (PHOTO-PAINT)

.EffectSoften .Percentage = *long*

This command smoothes and tones down harsh edges with only minimal loss of image detail.

Parameter	Description
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectSolarize (PHOTO-PAINT)

.EffectSolarize .Level = *long*

This command transforms colors to appear like those of a negative photographic image.

Parameter	Description
.Level	Lets you specify the intensity of the effect. Valid values range from 0 to 255.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectSwirl (PHOTO-PAINT)

.EffectSwirl .Angle = *long*, .CenterX = *long*, .CenterY = *long*

This command creates a swirling vortex of distortion on your image.

Parameter	Description
.Angle	Lets you specify the angle through which the swirl occurs. Values range from -3600 to 3600.
.CenterX	Lets you specify the horizontal position of the swirl's center.
.CenterY	Lets you specify the vertical position of the swirl's center.

Example

```
.EffectSwirl -719, 233, 121
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectTheBoss (PHOTO-PAINT)

.EffectTheBoss .Angle = *long*, .BevelWidth = *long*, .BevelHeight = *long*, .Brightness = *long*, .Direction = *long*, .Dropoff = *long*, .Sharpness = *long*, .Smooth = *long*, .Invert = *boolean*

This command applies The Boss effect to an image. You must create a mask before applying this effect.

Parameter	Description
.Angle	Lets you specify the angle at which the light bounces off the bevel.
.BevelWidth	Lets you specify the width of the bevel.
.BevelHeight	Lets you specify the height of the bevel.
.Brightness	Lets you specify the intensity of the light source.
.Direction	Lets you specify the angle at which the light hits the bevel.
.Dropoff	Lets you specify the edges style of the bevel.
.Sharpness	Lets you specify the amount of fading at the edge of the light shaft. A lower value results in a concentrated light source (like a flashlight). A higher value results in a softer, larger light source (like a ceiling light).
.Smooth	Lets you specify the smoothness of the bevel.
.Invert	0 does not invert the mask 1 inverts the mask

Example

```
.EffectTheBoss 45, 20, 30, 60, 135, 0, 0, 30
```

This example applies The Boss effect to an image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectTile (PHOTO-PAINT)

.EffectTile .Horizontal = *long*, .Vertical = *long*

This command reduces the dimensions of your image and reproduces the image as a series of tiles on a grid.

Parameter	Description
.Horizontal	Lets you specify the number of times the image appears along the horizontal axis.
.Vertical	Lets you specify the number of times the image appears along the vertical axis.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectTool (PHOTO-PAINT)

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

This command applies an effect stroke along a path defined by a series of StartDraw and ContinueDraw commands. An EffectTool command block must end with an EndDraw command.

Parameter	Description
.BrushID	Lets you specify the brush to use by index number.
.TypeID	Lets you specify the type of brush stroke by index number.
.MergeMode	Lets you specify 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	Lets you specify the rate at which the effect or paint is applied to the image, ranging from 1 to 100. A higher value results in a more pronounced effect or heavier application of paint.
.NibShape	Lets you specify the shape of the nib: 0 = Round 1 = Square
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Transparency	Lets you specify the transparency of the brush stroke. Valid values range from 0 to 99%.
.Rotate	Lets you specify the angle at which the nib is rotated. Valid values range from 0 to 360 degrees.
.Flatten	Lets you specify the flatness of the nib. Valid values range from 0 to 99.
.SoftEdge	Lets you specify the transparency of the nib's edges. As you increase the value of this setting, the soft edge expands to eventually reach the center of the stroke. Low values affect only the rim of the stroke. Valid values range from 0 to 100%.

Example

```
.EffectTool 7, 0, 0, 51, 0, 30, 0, 0, 100, 50
    .BrushTextureSettings "", 0, 0, 0, 0, 1, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 186, 159, 106, 0
    .RandomSeed 1691418496
    .StartDraw 27776, 12160, 0, 0
    .ContinueDraw 28779, 13945, 0, 0
    ...
    .ContinueDraw 36293, 64568, 0, 0
    .EndDraw
```

This example applies an effect stroke along the defined path.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectTraceContour (PHOTO-PAINT)

.EffectTraceContour .Level = *long*, .EdgeType = *long*

This command creates edges of different intensity by tracing image elements using the 16 colors of the standard VGA palette.

Parameter	Description
.Level	Lets you specify the brightness threshold for outlining. Valid values range from 1 to 255.
.EdgeType	Lets you specify the type of edges to trace: 0 = Traces the areas of your image where the brightness levels of the pixels fall below the .Level value 1 = Traces the areas of your image where the brightness level of the pixels exceeds the .Level value

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectUnsharpMask (PHOTO-PAINT)

.EffectUnsharpMask .Radius = *long*, .Percentage = *long, .*Threshold = *long*

This command accentuates edge detail as well as focusing some blurred areas in the image.

Parameter	Description
.Radius	Lets you specify the range of the effect. Larger values increase the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.
.Threshold	Lets you specify the change in value that is required by a pixel before the Unsharp Mask effect is applied.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectUserDefined (PHOTO-PAINT)

.EffectUserDefined .Divisor = *long*, .Offset = *long*, .Intensity = *boolean*

This command applies a custom convolution to the current image. An EffectUserDefined command block must contain an EffectUserDefinedPoint command for each of the 25 points in the convolution matrix, and must end with an EndEffectUserDefined command.

Parameter	Description
.Divisor	Lets you specify the divisor value. After the command multiplies each matrix value by the brightness value of the corresponding pixel, it adds the products together, and then divides the sum by the value you type in the Divisor box.
.Offset	Lets you specify the offset value. This is the value that will be added to the final pixel values just before the effect is applied.
.Intensity	0 sharpens all values 1 sharpens only intensity

Note

.EffectUserDefined starts the command block, .EffectUserDefinedPoint fills the user array, and .EndEffectUserDefined ends the command.

Example

```
.EffectUserDefined 1, 0
    .EffectUserDefinedPoint 0, 1
    .EffectUserDefinedPoint 1, 1
    .EffectUserDefinedPoint 2, 1
    .EffectUserDefinedPoint 3, 1
    ...
    .EffectUserDefinedPoint 24, 1
    .EndEffectUserDefined
```

This example fills the user array with values of 1, which will give the effect of smoothing the image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectUserDefinedPoint (PHOTO-PAINT)

.EffectUserDefinedPoint .Index = *long*, .Value = *long*

This command defines a point in the convolution matrix of an EffectUserDefined command. An EffectUserDefined command block must contain an EffectUserDefinedPoint command for each of the 25 points in the convolution matrix.

Parameter	Description
.Index	Lets you specify the index of the matrix point to define. Valid values range from 0 to 24.
.Value	Lets you specify the value of the specified matrix entry. Valid values range from -999 to 999.

Note

.EffectUserDefined starts the command block, .EffectUserDefinedPoint fills the user array, and .EndEffectUserDefined ends the command.

Example

```
.EffectUserDefined 1, 0
    .EffectUserDefinedPoint 0, 1
    .EffectUserDefinedPoint 1, 1
    .EffectUserDefinedPoint 2, 1
    .EffectUserDefinedPoint 3, 1
    ...
    .EffectUserDefinedPoint 24, 1
    .EndEffectUserDefined
```

This example fills the user array with values of 1, which will give the effect of smoothing the image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectVignette (PHOTO-PAINT)

.EffectVignette .Shape = *long*, .Offset = *long*, .Fade = *long, .*ColorModel = *long, .*Color1 = *long, .*Color2 = *long, .*Color3 = *long, .*Color4 = *long*

This command creates a frame around your image.

Parameter	Description
.Shape	Lets you specify the frame shape: 0 = Elliptical 1 = Circular 2 = Rectangular 3 = Square
.Offset	Lets you specify the size of the frame. Valid values range from 0 to 140.
.Fade	Lets you specify the fade rate between the image and the frame. Valid values range from 0 to 100.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectVignette 1, 109, 33, 5, 0, 255, 0, 0
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectWetPaint (PHOTO-PAINT)

.EffectWetPaint .Wetness = *long*, .Percentage = *long*

This command creates the illusion that your image is a painting that is still wet. The effects can range from subtle changes in the luminescence of colors to streaks of wet paint dripping down your image

Parameter	Description
.Wetness	Lets you specify the range of colors that drip. Negative values cause the dark colors to drip, positive values cause the light colors to drip. Valid values range from -50 to 50.
.Percentage	Lets you specify the size of the paint drip. Valid values range from 0 to 100%.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectWhirlpool (PHOTO-PAINT)

.EffectWhirlpool .Spacing = *long*, .Smear = *long*, .Twist = *long*, .Streak = *long*, .Warp = *boolean*

This command applies a pattern of fluid streamlines over your image.

Parameter	Description
.Spacing	Lets you specify the spacing between swirls. Valid values range from 5 to 200.
.Smear	Lets you specify the length of the swirls. Valid values range from 3 to 30.
.Twist	Lets you specify the degree of curvature. Valid values range from 0 to 90.
.Streak	Lets you specify the intensity of the swirls. Valid values range from 0 to 100.
.Warp	Set to TRUE (-1) if you want the whirlpool effect to distort the image. Set to FALSE (0) if you want the effect to overlay the image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EffectWind (PHOTO-PAINT)

.EffectWind .Strength = *long*, .Opacity = *long*, .Direction = *long*

This command blurs your image in a specific direction, creating the effect of wind blowing across your image.

Parameter	Description
.Strength	Lets you specify the intensity of the effect. Valid values range from 0 to 100%.
.Opacity	Lets you specify the opacity of the effect. Valid values range from 1 to 100%.
.Direction	Lets you specify the direction of the blur. Valid values range from 0 to 360 degrees.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EffectZigZag (PHOTO-PAINT)

.EffectZigZag .Period = *long*, .Strength = *long*, .Damping = *long*, .Type = *long,* .CenterX = *long,* .CenterY = *long*

This command distorts an image by bending the image lines that run from the center of the image to its edge. This effect produces waves of straight lines and angles which seem to twist the image from its center outwards.

Parameter	Description
.Period	Lets you specify the period of the waves. Valid values range from 1 to 100.
.Strength	Lets you specify the intensity of the distortion. Valid values range from 1 to 100.
.Damping	Lets you specify the degree of damping in successive waves. Large values cause the distortion waves to phase out toward the edges of your image, Smaller values cause the waves to extend toward the edges. Valid values range from 1 to 100.
.Type	Lets you specify the type of wave: 0 = Pond ripples 1 = Out from center 2 = Around center
.CenterX	Lets you specify the horizontal position of the zig zag effect.
.CenterY	Lets you specify the vertical position of the zig zag effect.

Example

```
.EffectZigZag 3, -73, 20, 1, 99, 100
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndAdjustEffect (PHOTO-PAINT)

.EndAdjustEffect

This command ends an AdjustEffectInfo command block.

Example

```
.EffectAdjustBlur
    .AdjustEffectInfo 4, 10, 0, 0
    .AdjustEffectInfo 20, 10, 0, 0
    .AdjustEffectInfo 20, 10, 69, 0
    .AdjustEffectInfo 1, 10, 69, 0
    .EndAdjustEffect
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndColorEffect (PHOTO-PAINT)

.EndColorEffect

This command ends nine effects commands whether they are applied to the image itself or to a lens object (lenses are only available in the Corel PHOTO-PAINT 7 Plus standalone program). The effect commands that use this command are: .ImageReplaceColors, .ImagePosterize, .ImageLevelThreshold, ImageInvert, ImageColorBalance, .ImageBCI, .ImageHSL, .ImageDesaturate, and .ImageGamma.

Example

```
.ImageLevelThreshold 0, 179, 255, 255,0
    .EndColorEffect
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndEffectLighting (PHOTO-PAINT)

.EndEffectLighting

This command ends an EffectLighting command block.

Example

```
.EffectLighting 4
    .EffectLightSource 0, 0.5, 0.5, 0, 75, 180, 90, 40, 180, 100, 0, 100, 0, 0, 0, 255, 0, 0
    .EffectLightSource 1, 0.097166, 1.09717, 1, 60, 15, 25, 95, 45, 80, 0, 100, 0, 0, 0, 255,
255, 51
    .EffectLightSource 2, 0.433, 1.1, 1, 60, 15, 25, 95, 45, 80, 0, 100, 0, 0, 0, 204, 102, 255
    .EffectLightSource 3, 0.766, 1.1, 1, 60, 15, 25, 95, 45, 80, 0, 100, 0, 0, 0, 255, 255, 255
    .EndEffectLighting
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndEffectMeshWarp (PHOTO-PAINT)

.EndEffectMeshWarp

This command ends an EffectMeshWarp command block.

Example

```
.EffectMeshWarp 5, 5
    .EffectMeshPoint 0, 0, 0, 0
    .EffectMeshPoint 0, 1, 0.25, 0
    .EffectMeshPoint 0, 2, 0.5, 0
    .EffectMeshPoint 0, 3, 0.75, 0
    .EffectMeshPoint 0, 4, 1, 0
    .EffectMeshPoint 1, 0, 0, 0.25
    .EffectMeshPoint 1, 1, 0.340067, 0.345946
    .EffectMeshPoint 1, 2, 0.52862, 0.297297
    .EffectMeshPoint 1, 3, 0.861953, 0.145946
    .EffectMeshPoint 1, 4, 1, 0.25
    .EffectMeshPoint 2, 0, 0, 0.5
    .EffectMeshPoint 2, 1, 0.25, 0.5
    .EffectMeshPoint 2, 2, 0.579125, 0.556757
    .EffectMeshPoint 2, 3, 0.75, 0.5
    .EffectMeshPoint 2, 4, 1, 0.5
    .EffectMeshPoint 3, 0, 0, 0.75
    .EffectMeshPoint 3, 1, 0.121212, 0.859459
    .EffectMeshPoint 3, 2, 0.5, 0.75
    .EffectMeshPoint 3, 3, 0.818182, 0.881081
    .EffectMeshPoint 3, 4, 1, 0.75
    .EffectMeshPoint 4, 0, 0, 1
    .EffectMeshPoint 4, 1, 0.25, 1
    .EffectMeshPoint 4, 2, 0.5, 1
    .EffectMeshPoint 4, 3, 0.75, 1
    .EffectMeshPoint 4, 4, 1, 1
    .EndEffectMeshWarp
```

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

EndEffectShear (PHOTO-PAINT)

.EndEffectShear

This command ends an EffectShear command block.

Example

```
.EffectShear 50, 0, 1
    .EffectShearTable 0, 264
    .EffectShearTable 1, 265
    .EffectShearTable 2, 267
    .EffectShearTable 3, 269
    ...
    .EffectShearTable 1021, 779
    .EffectShearTable 1022, 780
    .EffectShearTable 1023, 781
    .EndEffectShear
```

This example applies a shear effect to the active image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

EndEffectUserDefined (PHOTO-PAINT)

.EndEffectUserDefined

This command ends an EffectUserDefined command block.

Note

.EffectUserDefined starts the command block, .EffectUserDefinedPoint fills the user array, and .EndEffectUserDefined ends the command.

Example

```
.EffectUserDefined 1, 0
    .EffectUserDefinedPoint 0, 1
    .EffectUserDefinedPoint 1, 1
    .EffectUserDefinedPoint 2, 1
    .EffectUserDefinedPoint 3, 1
    ...
    .EffectUserDefinedPoint 24, 1
    .EndEffectUserDefined
```

This example fills the user array with values of 1, which will give the effect of smoothing the image.

{button ,AL(`OVR1 Effect commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

Path commands (PHOTO-PAINT)

PathCreate (PHOTO-PAINT)

.PathCreate .Nodes = *long*

This command creates a new path using nodes defined by a series of PathNode commands. A PathCreate command block must end with a PathEnd command.

Parameter	Description
.Nodes	Lets you specify the number of nodes used to define the path.

Example

```
.PathCreate 10
    .PathNode 36, 32, FALSE, 0, 0
    .PathNode 36, 32, FALSE, 0, 3
    .PathNode 15, 86, FALSE, 0, 3
    .PathNode 88, 118, FALSE, 2, 2
    .PathNode 161, 150, FALSE, 1, 3
    .PathNode 110, 108, FALSE, 1, 3
    .PathNode 179, 136, FALSE, 2, 2
    .PathNode 248, 164, FALSE, 1, 3
    .PathNode 237, 219, FALSE, 1, 3
    .PathNode 237, 219, FALSE, 1, 2
    .PathEnd
```

This example creates a ten-node path.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathCreateFromMask (PHOTO-PAINT)

.PathCreateFromMask .Tightness = *long*, .Threshold = *long*

This command creates a path that has the shape of the current mask marquee.

Parameter	Description
.Tightness	Lets you specify the number of nodes for the new path.
.Threshold	Lets you specify the angle size required between sections of the selection's boundary for a node top be placed at the intersection of the sections.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathDelete (PHOTO-PAINT)

.PathDelete .FromDisk = *boolean*

This command deletes the selected path.

Parameter	Description
.FromDisk	Set to TRUE (-1) to also delete the file containing a saved path.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

PathEnd (PHOTO-PAINT)

.PathEnd

This command ends a PathCreate command block.

Example

```
.PathCreate 10
    .PathNode 36, 32, FALSE, 0, 0
    .PathNode 36, 32, FALSE, 0, 3
    .PathNode 15, 86, FALSE, 0, 3
    .PathNode 88, 118, FALSE, 2, 2
    .PathNode 161, 150, FALSE, 1, 3
    .PathNode 110, 108, FALSE, 1, 3
    .PathNode 179, 136, FALSE, 2, 2
    .PathNode 248, 164, FALSE, 1, 3
    .PathNode 237, 219, FALSE, 1, 3
    .PathNode 237, 219, FALSE, 1, 2
    .PathEnd
```

This example creates a ten-node path.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathImportVector (PHOTO-PAINT)

.PathImportVector .FileName = *string,* .ScaleX = *double,* .ScaleY = *double*

This command imports vector images as paths. This command can only load vector file formats.

Parameter	Description
.FileName	Lets you specify the name of the vector to import.
.ScaleX	Lets you specify the horizontal scaling factor. A vector image can be scaled to any size on import.
.ScaleY	Lets you specify the vertical scaling factor. A vector image can be scaled to any size on import.

Example

```
.PathImportVector sample.cdr 1.5, 2
```

This example imports a CorelDRAW vector image named "sample" as a path.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} Related Topics

PathLoad (PHOTO-PAINT)

.PathLoad .PathName = *string,* ScaleX = *double,* ScaleY = *double*

This command opens a path that has been saved to disk.

Parameter	Description
.PathName	The name of the file containing the saved path.
.ScaleX	Lets you specify the horizontal scaling factor. A vector image can be scaled to any size on import.
.ScaleY	Lets you specify the vertical scaling factor. A vector image can be scaled to any size on import.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathNew (PHOTO-PAINT)

.PathNew

This command deletes the existing path so that you may create a new one.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathNode (PHOTO-PAINT)

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

This command defines a single node in a path for the PatchCreate command.

Parameter	Description
.ptX	Lets you specify the horizontal coordinate of the node point.
.ptY	Lets you specify the vertical coordinate of the node point.
.Closed	Set to TRUE (-1) if you want the current point to close the path.
.Continuity	Lets you specify how the curve is derived: 0 = First order integral 1 = Second order integral
.Type	Lets you specify the node type: 0 = Line 1 = Curve 2 = Normal 3 = Control point

Example

```
.PathCreate 10
    .PathNode 36, 32, FALSE, 0, 0
    .PathNode 36, 32, FALSE, 0, 3
    .PathNode 15, 86, FALSE, 0, 3
    .PathNode 88, 118, FALSE, 2, 2
    .PathNode 161, 150, FALSE, 1, 3
    .PathNode 110, 108, FALSE, 1, 3
    .PathNode 179, 136, FALSE, 2, 2
    .PathNode 248, 164, FALSE, 1, 3
    .PathNode 237, 219, FALSE, 1, 3
    .PathNode 237, 219, FALSE, 1, 2
    .PathEnd
```

This example creates a ten-node path.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathSave (PHOTO-PAINT)

.PathSave .PathName = *string*

This command saves the existing path to disk so that you may use it in the future in any image. Paths are given a .PTH file extension.

Parameter	Description
.PathName	Lets you specify the filename of the path you want to save.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

PathStroke (PHOTO-PAINT)

.PathStroke

This command ends any of the CloneTool, BrushTool, or EffectTool command blocks.

Example

```
.BrushTool 0, 0, 0, 20, 0, 5, 0, 0, 100, 0
    .BrushTextureSettings "", 0, 0, 0, 0, 0, TRUE, FALSE
    .BrushDabSettings 1, 25, 0, 0, 0, 0, 0
    .SetPaintColor 5, 186, 159, 106, 0
    .PathStroke
```

This example applies a brush stroke to the current image.

{button ,AL(`OVR1 Path commands PHOTOPAINT;',0,"Defaultoverview",)} **Related Topics**

MaskFloaterTranslate (PHOTO-PAINT)

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

This command lets you move a mask floater.

Parameter	Description
.Left	The number of pixels, from the left, to move the mask floater
.Bottom	The number of pixels, from the center, to move the mask floater
.Copy	0 cuts the mask floater 1 copies the mask floater

GuidelineSelect (PHOTO-PAINT)

.GuidelineSelect .GuideID = *long*, .Selected = *boolean*

This command lets you select a guideline.

Parameter	Description
.GuideID	The index of the guideline you want to select. The index of the first guideline you created is 1, the second is 2, etc.,...
.Selected	0 does not select the guideline 1 selects the guideline

ImageApplyICCProfile (PHOTO-PAINT)

.ImageApplyICCProfile .Filename = *string*

This command applies a color correction profile to an image.

Parameter	Description
.Filename	The filename of the color correction profile.

EndGuideline (PHOTO-PAINT)

.EndGuideline

This command ends a block of guideline commands.

GuidelineAdd (PHOTO-PAINT)

.GuidelineAdd .X = *long*, .Y = *long*, .Angle = *long*, .Horizontal = *boolean*

This command adds a guideline.

Parameter	Description
.X	The x coordinate
.Y	The y coordinate
.Angle	The angle, in degrees, at which to add the guideline. The default is 0 degrees.
.Horizontal	0 adds a vertical guideline 1 adds a horizontal guideline

GuidelineMove (PHOTO-PAINT)

.Move .X - *long*, .Y = *long*

This command moves a guideline

Parameter	Description
.X	The x coordinate
.Y	The y coordinate

GuidelineDelete (PHOTO-PAINT)

.GuidelineDelete

This command deletes a guideline.

EndObject (PHOTO-PAINT)

.EndObject

This command ends a block of object commands.

EndColorReplace (PHOTO-PAINT)

.EndColorReplace

This command ends a block of color replace commands.

GetDocumentHRes (PHOTO-PAINT)

.GetDocumentHRes

This command returns the horizontal resolution of a document.

GetDocumentVRes (PHOTO-PAINT)

.GetDocumentVRes

This command returns the vertical resolution of a document.

GetDocumentXRulerUnits (PHOTO-PAINT)

.GetDocumentXRulerUnits

This command returns the ruler units for the x-axis.

GetDocumentYRulerUnits (PHOTO-PAINT)

.GetDocumentYRulerUnits

This command returns the ruler units for the y axis.

GetDocumentXGridFrequency (PHOTO-PAINT)

.GetDocumentXGridFrequency

This command returns the frequency of the grid for the x axis.

GetDocumentYGridFrequency (PHOTO-PAINT)

.GetDocumentYGridFrequency

This command returns the frequency of the grid for the y axis.

EffectBumpMap (PHOTO-PAINT)

.EffectBumpMap .Filename = *string*, .Mode = *boolean*, .TileWidth = *long*, .TileHeight = *long,* .Floor = *long*, .Ceiling = *long*, .ScaleFactor = *long*, .Highlight = *long*, .Invert - *boolean*, .Smooth = *boolean*, .AmbienBrightness = *long*, .AmbientColorModel = *long*, .AmbientColor1 = *long*, .AmbientColor2 = *long*, .AmbientColor3 = *long*, .AmbientColor4 = *long*, .DirectionalBrightness = *long*, .DirectionalColorModel = *long*, .DirectionalColor1 = *long*, .DirectionalColor2 = *long*, .DirectionalColor3 = *long*,. DirectionalColor4 = *long*, .Direction = *long*, .Declination = *long*

This command applies the bump map effect.

Parameter	Description
.Filename	Lets you specify the name of the file.
.Mode	0 stretches the image 1 tiles the bumpmap to fit the image.
.TileWidth	Lets you specify the width of each tile to stretch the bumpmap. 0 will automatically use the width of the bumpmap.
.TileHeight	Lets you specify the height of each tile to stretch the bumpmap. 0 will automatically use the height of the bumpmap.
.Floor	Lets you specify the minimum value allowed in the bumpmap. All values below this are clipped. Valid values range from 0 to 255.
.Ceiling	Lets you specify the maximum value allowed in the bumpmap. All values above this are clipped. Valid values range from 0 to 255.
.ScaleFactor	Lets you specify the percentage by which to scale the values in the bumpmap. Valid values range from 0 to 200.
.Highlight	Lets you specify the amount of specular highlight. Low values simulate a dull surface. High values simulate a shiny surface. Valid values range from 0 to 100.
.Invert	0 does not invert the bumpmap values so that dark values become bright and bright values become dark. 1 inverts the bumpmap values so that dark values become light and bright values become dark
.Smooth	0 Does not blur the bumpmap 1 blurs the bumpmap, resulting in a bumpmap with more unique values than before
.AmbientBrightness	Lets you specify the brightness of the ambient light, which comes from all directions and illuminates all pixels equally, regardless of their surface gradients. Valid values range from 0 to 100.
.AmbientColorModel	Lets you specify the color model to use for the ambient light. 3 CMYK 5 RGB 9 Grayscale
.AmbientColor1	First color component value. Valid values range from 0 to 255.
.AmbientColor2	Second color component value. Valid values range from 0 to 255.
.AmbientColor3	Third color component value. Valid values range from 0 to 255.
.AmbientColor4	Fourth color component value. Valid values range from 0 to 255.
.DirectionalBrightness	Lets you specify the brightness of the directional light, which illuminates each pixel by an amount dependent on its surface gradient. Valid values range from 0 to 100.
.DirectionalColorModel	Lets you specify the color model to use for the direction light. 3 CMYK 5 RGB 9 Grayscale
.DirectionalColor1	First color component value. Valid values range from 0 to 255.
.DirectionalColor2	Second color component value. Valid values range from 0 to 255.
.DirectionalColor3	Third color component value. Valid values range from 0 to 255.
.DirectionalColor4	Fourth color component value. Valid values range from 0 to 255.
.Direction	Lets you specify the angle, clockwise, describing the direction the directional light is travelling in the plane of the image. Valid values range from 0 to 359.
.Declination	Lets you specify the angle, clockwise, describing the direction the directional light is travelling. Valid values range from 0 to 359.

EffectAlchemy (PHOTO-PAINT)

.EffectAlchemy .Filename = *string*, .Layer = *long*, .Density = *long*, .Rand = *long*, .HPosV = *long*, .VPosV = *long*, .BrushClr = *boolean*, .BackClr = *boolean*, .ClrModel = *long*, .ClrP1 = *long*, .ClrP2 = *long*, .ClrP3 = *long*, .ClrP4 = *long*, .ClrMode2 = *long*, .ClrP5 = *long*, .ClrP6 = *long*, .ClrP7 = *long*, .ClrP8 = *long*, .HueV = *long*, .SatV = *long*, .BrtV = *long*, .AngleV = *long*, .BrushTrans = *long*, .Trans1 = *long*, .Trans2 = *long*, .TransV = *long*, .Dx = *double*, .Dy = *double*

The command applies the alchemy effect.

Parameter	Description
.Filename	
.Layer	
.Density	
.Rand	
.HPosV	
.VPosV	
.BrushClr	
.BackClr	
.ClrModel	
.ClrP1	
.ClrP2	
.ClrP3	
.ClrP4	
.ClrMode2	
.ClrP5	
.ClrP6	
.ClrP7	
.ClrP8	
.HueV	
.SatV	
.BrtV	
.AngleV	
.BrushTrans	
.Trans1	
.Trans2	
.TransV	
.Dx	
.Dy	

EffectBrushStroke (PHOTO-PAINT)

.EffectBrushStroke .Distribution = *long*, .BrushShape = *long*

This command applies the brush stroke effect.

Parameter	Description
.Distribution	
.BrushShape	

EffectCrystalize (PHOTO-PAINT)

.EffectCrystalize .Size = *long*

This command applies the crystalize effect

Parameter	Description
.Size	Lets you specify the size of the crystals. Valid values range from 1 to 100 percent. The value represents the approximate percentage of the crystal size to the image size.

Example

```
.EffectCystalize 10
```

EffectKidsPlay (PHOTO-PAINT)

.EffectKidsPlay .Percent = *long*, .Order = *long*, .Brightness = *long*, .Size = *long*, .Angle = *long*

This command applies the kids play effect.

Parameter	Description
.Percent	
.Order	
.Brightness	
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Angle	

EffectWavePaper (PHOTO-PAINT)

.EffectWavePaper .pressure = *long*, .ColorMode = *long*

This command applies the wave paper effect.

Parameter	Description
.pressure	
.ColorMode	

EffectFabric (PHOTO-PAINT)

.EffectFabric .Style = *long*, .Percent = *long*, .Brightness = *long*, .Size = *long*, .Angle = *long*

This command applies the fabric effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Percent	
.Brightness	
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Angle	

EffectConte (PHOTO-PAINT)

.EffectConte .Intensity = *long*, .Texture = *long*

This command applies the conte effect.

Parameter	Description
.Intensity	
.Texture	

EffectCraft (PHOTO-PAINT)

.EffectCraft .Brightness = *long*, .Size = *long*, .Style = *long*, .Percent = *long*, .Angle = *long*

This command applies the craft effect.

Parameter	Description
.Brightness	
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Percent	
.Angle	

EffectSkin (PHOTO-PAINT)

.EffectSkin .Age = *long*, .Variation = *long*, .Type = *long*, .Comp1 = *long*, .Comp2 = *long*, .Comp3 = *long*, .Comp4 = *long*

This command applies the skin effect.

Parameter	Description
.Age	
.Variation	
.Type	
.Comp1	
.Comp2	
.Comp3	
.Comp4	

EffectWaterMarker (PHOTO-PAINT)

.EffectWaterMarker .Mode = *long*, .Color = *long*

This command applies the water marker effect.

Parameter	Description
.Mode	
.Color	

EffectPastel (PHOTO-PAINT)

.EffectPastel .Mode = *long*, .Size = *long*, .Hue = *long*

This command applies the pastel effect.

Parameter	Description
.Mode	
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Hue	

EffectPencil (PHOTO-PAINT)

.EffectPencil .Mode = *long*, .Style = *long*, .Edge = *long*, .Bright = *long*

This command applies the pencil effect.

Parameter	Description
.Mode	
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Edge	
.Bright	

EffectCharcoal (PHOTO-PAINT)

.EffectCharcoal .Size = *long*, .Edge = *long*

This command applies the charcoal effect.

Parameter	Description
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Edge	

EffectCrayon (PHOTO-PAINT)

.EffectCrayon .Detail =-*long*, .Edge = *long*

This command applies the crayon effect.

Parameter	Description
.Detail	
.Edge	

EffectMonet (PHOTO-PAINT)

.EffectMonet .Style = *long*, .Detail = *long*, .Hue = *long*, .Brightness = *long*

This command applies the monet effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Detail	
.Hue	
.Brightness	

EffectPlaster (PHOTO-PAINT)

.EffectPlaster .Detail = *long*, .Variation = *long*, .Brightness = *long*

This command applies the plaster effect.

Parameter	Description
.Detail	
.Variation	
.Brightness	

EffectPointillist (PHOTO-PAINT)

.EffectPointillist .Detail = *long*, .Bright = *long*, .Type = *long*, Comp1 - *long*, .Comp2 = *long*, .Comp3 = *long*, .Comp4 = *long*

This command applies the pointillist effect.

Parameter	Description
.Detail	
.Bright	
.Type	
.Comp1	
.Comp2	
.Comp3	
.Comp4	

EffectCubist (PHOTO-PAINT)

.EffectCubist .Detail = *long*, .Bright = *long*, .Type = *long*, .Comp1 = *long*, .Comp2 = *long*, .Comp3 = *long*, .Comp4 = *long*

This command applies the cubist effect.

Parameter	Description
.Detail	
.Bright	
.Type	
.Comp1	
.Comp2	
.Comp3	
.Comp4	

EffectVortex (PHOTO-PAINT)

.EffectVortex .Style = *long*, .Inner = *long*, .Outer = *long*, .XCenter = *long*, .YCenter = *long*

This command applies the vortex effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Inner	
.Outer	
.XCenter	
.YCenter	

EffectMosaic (PHOTO-PAINT)

.EffectMosaic .Style = *long*, .Size = *long*, .ColorMode = *long*, .Type = *long*, .Comp1 = *long*, .Comp2 = *long*, .Comp3 = *long*, .Comp4 = *long*

This command applies the mosaic effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.ColorMode	
.Type	
.Comp1	
.Comp2	
.Comp3	
.Comp4	

EffectPaletteKnife (PHOTO-PAINT)

.EffectPaletteKnife .Size = *long*, .Soft = *long*, .Angle = *long*

This command applies the palette knife effect.

Parameter	Description
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Soft	
.Angle	

EffectScratchPaper (PHOTO-PAINT)

.EffectScratchPaper .Style = *long*, .Density = *long*, .Size = *long*

This command applies the scratch paper effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Density	
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.

EffectStainedGlass (PHOTO-PAINT)

.EffectStainedGlass .Size = *long*, .Thickness = *long*, .Lighting = *boolean*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command applies the stained glass effect.

Parameter	Description
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Thickness	
.Lighting	
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

EffectScreenDoor (PHOTO-PAINT)

.EffectScreenDoor .Color = *long*, .Density = *long*, .Softness = *long*, .Brightness = *long*

This command applies the screen door effect.

Parameter	Description
.Color	
.Density	
.Softness	
.Brightness	

EffectUnderpaint (PHOTO-PAINT)

.EffectUnderpaint .Size = *long*, .Brightness = *long*

This command applies the underpaint effect.

Parameter	Description
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Brightness	

EffectAtmosphere (PHOTO-PAINT)

.EffectAtmosphere .Style = *long*, .Strength = *long*, .Size = *long*, .Variation = *long*, .Angle = *long*

This command applies the atmosphere effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Strength	
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Variation	
.Angle	

EffectWatercolor (PHOTO-PAINT)

.EffectWatercolor .Size = *long*, .Bleed = *long*, .PaperDetail = *long*, .Water = *long*, .Brightness = *long*

This command applies the watercolor effect.

Parameter	Description
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Bleed	
.PaperDetail	
.Water	
.Brightness	

EffectInkpen (PHOTO-PAINT)

.EffectInkpen .Mode = *long*, .Density = *long*, .Inkpool = *long*

This command applies the inkpen effect.

Parameter	Description
.Mode	
.Density	
.Inkpool	

EffectLocalHistogram (PHOTO-PAINT)

.EffectLocalHistogram .Width = *long*, .Height = *long*

This command applies the local histogram effect. EffectLocalHistogram enhances local contrast to reveal details in both light and dark areas.

Parameter	Description
.Width	Lets you specify the width of the region surrounding each pixel in which a histogram is generated. Valid values range from 5 to 255.
.Height	Lets you specify the height of the region surrounding each pixel in which a historgram is generated. Valid values range from 5 to 255.

Example

```
.EffectLocalHistogram 5, 255
```

EffectEtching (PHOTO-PAINT)

.EffectEtching .Detail = long, .Depth = long, .Direction = long, .ColorModel = long, .Color1 = long, .Color2 = long, .Color3 = long, .Color4 = long

This command applies the etching effect.

Parameter	Description
.Detail	Lets you specify the amount of detail in the etching. Lower values result in more flat areas. Valid values range from 1 to 100.
.Depth	Lets you specify the depth of the etching. Lower values result in more shallow surface relief. Valid values range from 1 to 100.
.Direction	Lets you specify the angle, clockwise, describing the direction the light is travelling in the plane of the image. Valid values range from 0 to 359 degrees.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectEtching 15, 1, 300, 5, 100, 100, 100, 0
```

EffectReliefSculpture (PHOTO-PAINT)

.EffectReliefSculpture .Detail =-*long*,

This command applies the relief sculpture effect. EffectReliefDetail makes the image look like a relief sculpture.

Parameter	Description
.Detail	Lets you specify the amount of detail in the relief sculpture. Lower values result in more flat areas. Valid values range from 1 to 100.

Example

```
.EffectReliefSculpture 55
```

EffectPlastic (PHOTO-PAINT)

.EffectPlasticWrap .Highlight = *long*, .Depth = *long*, .Smoothness = *long*, .Direction = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command applies the plastic wrap effect. EffectPlastic makes the image look like it is composed of plastic, giving it three-dimensional shadowing and highlights.

Parameter	Description
.Highlight	Lets you specify the amount of specular light. Low values simulate a dull surface. High values simulate a shiny surface. Valid values range from 0 to 100.
.Depth	Lets you specify the depth of the plastic. Lower values result in more shallow surface relief. Valid values range from 1 to 100.
.Smoothness	Lets you specify the smoothness of the plastic. Valid values range from 0 to 100.
.Direction	Lets you specify the angle, clockwise, describing the direction that the light is travelling. Valid values range from 0 to 359.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.EffectPlastic 10, 44, 23, 200, 5, 50, 50, 50, 0
```

EffectStone (PHOTO-PAINT)

.EffectStone .Minimum = *long*, .Amount = *long*, .Density = *long*, .Radius = *long*, .Invert = *boolean*, .Direction = *long*

This command applies the stone effect.

Parameter	Description
.Minimum	
.Amount	
.Density	
.Radius	
.Invert	
.Direction	

EffectBrickWall (PHOTO-PAINT)

.EffectBrick .Roughness = *long*, .BrickWidth = *long*, .BrickHeight = *long*,, .GroutWidth = *long*, .Direction = *long*

This command applies the brick effect. EffectBrickWall makes the image look like it is painted on a brick wall.

Parameter	Description
.Roughness	Lets you specify the roughness of the brick surface. Valid values range from 0 to 100.
.BrickWidth	Lets you specify the width, in pixels, of each brick. Valid values range from 1 to 1000.
.BrickHeight	Lets you specify the height, in pixels, of each brick. Valid values range from 1 to 1000.
.GroutWidth	Lets you specify the width, in pixels, of the grouts. Valid values range from 1 go 100.
.Direction	Lets you specify the angle, clockwise, describing the direction the light travels in the plane of the image. Valid values range from 0 to 359.

Example

```
.EffectBrickWall 30, 900, 450, 35, 300
```

EffectBubbles (PHOTO-PAINT)

.EffectBubbles .Diameter = *long*, .Coverage = *long*, .Direction = *long*, .Refraction = *boolean*, .Seed = *long*

This command applies the bubbles effect. EffectBubbles makes the image look like it has soap bubbles on it.

Parameter	Description
.Diameter	Lets you specify the average diameter of the bubbles. Valid values range from 20 to 400.
.Coverage	Lets you specify the approximate percentage of the image area to cover with bubbles. Valid values range from 1 to 100.
.Direction	Lets you specify the angle, clockwise, describing the direction the light travels in the plane of the image. Valid values range from 0 to 359.
.Refraction	0 does not distort the image to simulate the refraction of light passing through the bubbles. 1 distorts the image to simulate the refraction of light passing through the bubbles.
.Seed	Lets you specify the seed for randomizing the size and position of the bubbles.

Example

```
.EffectBubbles 21, 76, 324, 0, 4
```

GetPixelColor (PHOTO-PAINT)

.GetPixelColor .X = *long*, .Y = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command returns the pixel color.

Parameter	Description
.X	Lets you specify the X coordinate.
.Y	Lets you specify the Y coordinate.
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.GetPixelColor 10, 10, 5, 34, 36, 222, 0
```

EffectParticle (PHOTO-PAINT)

.EffectParticle .Style = *long*, .Size = *long*, .Density = *long*, .HueAdj = *long*, .Trans = *long*, .Angle = *long*

This command applies the particle effect.

Parameter	Description
.Style	Lets you specify the type of image to create: 0 = Monotone 1 = Duotone 2 = Tritone 3 = Quadtone
.Size	Lets you specify the width of the nib. Valid values range from 1 to 999 pixels.
.Density	
.HueAdj	
.Trans	
.Angle	

EffectCobblestone (PHOTO-PAINT)

.EffectCobblestone .Roughness = *long*, .Size = *long*, .GroutWidth = *long*, .Direction = *long*, .Warp = *boolean*

This command applies the cobblestone effect. EffectCobblestone makes the image look like it is painted on cobblestone.

Parameter	Description
.Roughness	Lets you specify the surface roughness of the cobblestones. Valid values range from 0 to 100.
.Size	Lets you specify the approximate diameter, in pixels, of the cobblestones. Valid values range from 0 to 100.
.GroutWidth	Lets you specify the width, in pixels, of the grouts. Valid values range from 0 to 100.
.Direction	Lets you specify the angle, clockwise, describing the direction the light is travelling in the plane of the image. Valid values range from 0 to 359.
.Warp	0 does not warp 1 warps the cobblestones to be more rounded.

Example

```
.EffectCobblestone 14, 10, 33, 120, 0
```

EffectFrame (PHOTO-PAINT)

.EffectFrame .Opacity = *long*, .Horizontal - *long*, .Vertical - *long*, .Fade = *long*, .HFlip = *long*, .VFlip = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*

This command applies the frame effect.

Parameter	Description
.Opacity	
.Horizontal	
.Vertical	
.Fade	
.HFlip	
.VFlip	
.ColorModel	Lets you specify the Color Model to use: 3 = CMYK (Cyan, Magenta, Yellow, Black) 5 = RGB (Red, Green, Blue) 8 = Black and White 9 = Grayscale
.Color1	Lets you specify the first color component for .ColorModel. For example, Red is the first color component for RGB.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Blue is the third color component for RGB. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. If this parameter is not available in the Color Model specified, set it to 0.

ObjectDropShadowDelete (PHOTO-PAINT)

.ObjectDropShadowDelete

This command deletes the drop shadow of an object

ObjectDropShadowCombine (PHOTO-PAINT)

.ObjectDropShadowCombine

This command combines the drop shadows of an object.

ObjectDropShadowSplit (PHOTO-PAINT)

.ObjectDropShadowSplit

This command splits the drop shadows of an object.

TextAppend (PHOTO-PAINT)

.TextAppend

This command appends text.

TextRender (PHOTO-PAINT)

.TextRender

This command ends a text block.

PathDuplicate (PHOTO-PAINT)

.PathDuplicate

This command duplicates a path.

ASC function

ASC(source)

Returns the numerical ANSI character value of the first character specified in a string. ASC is the opposite of the **CHR** function, which returns a character when the ANSI value is specified.

Parameter	Description
source	The string expression to be examined.

Note

- See the Corel SCRIPT Character Map for more ANSI details and Windows characters.

Example

```
i% = ASC("string")
```

This expression will assign the value 115, which is the ANSI value of the letter "s."

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

CHR function

CHR (value)

Returns the ANSI character that occupies the specified position in the ANSI character set. **CHR** is the opposite of **ASC**, which returns the ANSI code value when the character is entered. See the Corel SCRIPT Character Map for more ANSI details and Windows characters.

Parameter	Description
value	ANSI code value to be examined.

Example

```
s$ = CHR(65)
```

Assigns the letter "A" to the variable **s$**. (Character 65 of the ANSI character set is A.)

Special Characters

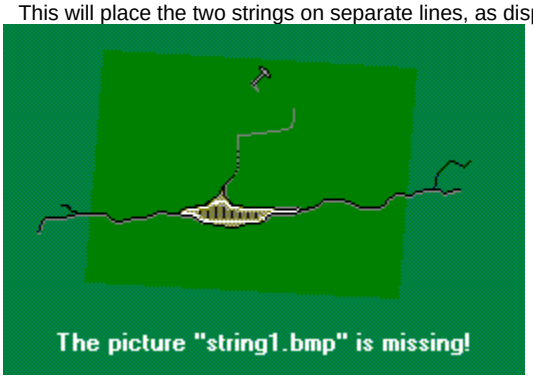
The CHR function is often used to add special characters to string variables that cannot be entered directly within double quotation marks. For example, to add double quotation marks to a string, you use character 34:

```
s$ = CHR(34) + "This will be in double quotes." + CHR(34)
MESSAGE s$
```

You can also use the function to add a return and a line feed within a string; use character 13 and 10, respectively:

```
s$ = "String 1" + CHR(13) + CHR(10) + "String 2"
MESSAGE s$
```

This will place the two strings on separate lines, as displayed in the message box.



The following table notes some of the special characters you can use with the **CHR** function.

Character Number	Special Character Returned
8	Backspace
9	Tab
10	Linefeed
13	Return
32	Space
34	Quotation mark

The following table notes some of the special characters you can use with the **CHR** function if you're using Corel VENTURA commands .**InsertSymbol** or .**TypeText**.

Character Number	Special Character Returned
10	Paragraph return
13	Forced line break
17	Em space
18	En space
19	Figure space
20	Thin space
21	Non-breaking space
22	Discretionary hyphen
34	Straight double quote

145	Typographical open single quote
146	Typographical close single quote
147	Typographical open double quote
148	Typographical close double quote
150	En dash
151	Em dash
153	Trademark
169	Copy right
174	Registered mark

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

INSTR function

INSTR (string1, string2, start**)**

Returns the starting position of the first occurrence of a string within another string. If the specified string is not found, the function returns 0.

Parameter	Description
string1	The string expression within which the search is made.
string2	The string for which you are searching.
start	Lets you specify the position where the search begins within **string1**. If unspecified, the search starts at the beginning of **string1** (same as **start**=1). Must be a non negative number and fractional numbers are rounded.

Example

```
pos = INSTR("Los Angeles", "Ang")
```

Sets **pos** to the value 5 because "Ang" occurs at the fifth character in the string "Los Angeles".

```
pos = INSTR("Los Angeles: City of Angels", "Ang", 8)
```

Sets **pos** to the value 22.

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

LCASE function

LCASE(source)

Converts a string to lowercase characters.

Parameter	Description
source	The string expression to convert.

Note

- Non letter characters do not change when this function is used.
- You can use **UCASE** to convert to uppercase characters.

Example

```
x$="HI"
firststring$ = LCASE(x)
secondstring$ = LCASE("THerE")
MESSAGE firststring + " " + secondstring
```

The above example displays the converted strings "hi there" in a message box.

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

LEFT function

LEFT (source, number)

Returns the specified number of characters from the beginning of a string.

Parameter	Description
source	The string expression from which the specified characters are returned.
number	Lets you specify the number of characters to be returned. Must be a non negative number and fractional numbers are rounded.

Note

- This function can be used to truncate user input from a dialog box.
- You can use the **LEN** function to determine the number of characters in a string.
- You can use the **RIGHT** function to return characters from the end of a string.

Example

```
abc$ = LEFT("I want to dance with you", 15)
MESSAGE abc$
```

Displays "I want to dance" in a message box.

Combine LEFT with INSTR to extract the portion of a string either up to or including a specified substring.

```
city$ = "San Francisco, California"
Mystr$ = LEFT(city$, INSTR(city$, ",")-1)
```

Extracts the characters in **city$** up to, but not including the comma, and places the result in the variable **Mystr$**. The variable **Mystr$** now has the value "San Francisco".

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

LEN function

LEN(source)

Returns the length or number of characters in a string.

Parameter	Description
source	The string expression that is measured.

Example

```
num = LEN("This is a test")
```

Assigns the length of the string, 14, to the variable **num**.

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

LTRIM function

LTRIM(source)

Removes any leading spaces from a string. You can use **LTRIM** to remove leading spaces from dialog box inputs.

Parameter	Description
source	The string expression from which leading spaces are removed.

Example

```
MyString$ = "   Test"
MyString$ = LTRIM(MyString$)
```

Assigns "Test" to the variable **MyString$**. All leading spaces that were previously in the variable are removed.

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

MID function and statement

Function: MID(source, index, count**)**
Statement: MID(source, index , count**) = modify**

If used as a function, MID returns a specified number of characters, starting at a specified position in a string. If used as a statement, MID replaces a portion of a string with another string, beginning at a specified character.

Parameter	Description
source	Any string, string variable, string constant, or expression returning a string. Hold the string to be modified.
source	For a function, the string expression from which to return characters. For a statement, the string expression holding the original string to be modified.
index	Position of the first character to be returned (function) or modified (statement).
count	For a function, the number of characters to be returned. For a statement, the number of characters to be overwritten. If not specified, the rest of **source** is returned or overwritten.
modify	A string expression replacing a portion of **source**.

Note

- You can use the **LEN** function to determine the number of characters in a string.

Example

```
s$ = MID("I want to dance with you", 11, 5)
```

The function extracts five characters from the string, beginning with the eleventh character. The variable **s$** then becomes "dance".

```
str1$ = "I want to dance with you"
MID(str1$, 22, 3) = "him"
```

The statement changes three characters, starting with the 22nd character, to the new string. The **str1$** variable now contains "I want to dance with him".

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

RIGHT function

RIGHT (source, number)

Returns the specified number of characters from the end of a string.

Parameter	Description
source	The string expression from which the specified characters are returned.
number	Lets you specify the number of characters to be returned. Must be a non negative number and fractional numbers are rounded.

Note

- This function can be used to truncate user input from a dialog box.
- You can use the **LEN** function to determine the number of characters in a string.
- You can use the **LEFT** function to return characters from the beginning of a string.

Example

```
abc$ = RIGHT("I don't want to dance", 13)
MESSAGE abc$
```

Displays "want to dance" in a message dialog box.

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

RTRIM function

RTRIM(source)

Removes any trailing spaces from a string. You can use RTRIM to remove trailing spaces from dialog box inputs.

Parameter	Description
source	The string expression from which trailing spaces are removed.

Example

```
MyString$ = "Test   "
MyString$ = RTRIM(MyString$)
```

Assigns "Test" to the variable **MyString$**. All trailing spaces that were previously in the variable are removed.

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

SPACE function

SPACE (num)

Returns a string that consists of a specified number of spaces (ANSI character number 32).

Parameter	Description
num	Lets you specify the number of spaces to be included in a string.

Note

- See the Corel SCRIPT Character Map for more ANSI details and Windows characters.

Example

```
Mystr$ = SPACE(4) + "Test" + SPACE(4)
```

Makes the string variable **Mystr$** equal to " Test ". (The string **Mystr$** now consists of 4 spaces, the word TEST and another 4 spaces).

```
x1 = "Corel"
x2 = "SCRIPT"
x3 = x1 + SPACE(1) + x2
```

Makes the string variable **x3$** equal to "Corel SCRIPT".

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

STR function

STR(num)

Returns a string representation of a numeric data type. The **STR** function is useful when you want to manipulate a number as a string.

Parameter	Description
num	Lets you specify the numeric expression that is returned as string representation.

Note

- If a positive number is converted, the **STR** function inserts a leading space before the first character. If a negative number is converted, the **STR** function inserts a negative sign before the first character.
- You can use only the period as a decimal separator with the **STR** function. If you're not using a period (**.**) as a decimal separator, the **CSTR** function can be used to convert a number to a string. Your Windows decimal settings are set in the Control Panel.
- If you're converting dates, they must be in the standard International format (yy/MM/dd hh:mm:ss).

Example

```
aInteger$ = STR(72)
aNonInteger$ = STR(.140166)
```

The first example assigns "72" to the variable **aInteger$**. The second example assigns "0.140166" to **aNonInteger$**.

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

UCASE function

UCASE(source)

Converts a string to uppercase characters.

Parameter	Description
source	The string expression to convert.

Note

- Non letter characters do not change when this function is used.
- You can use **LCASE** to convert to lowercase characters.

Example

```
x$="hI"
firststring$ = UCASE (x)
secondstring$ = UCASE("THerE")
MESSAGE firststring + " " + secondstring
```

Displays the converted strings "HI THERE" in a message dialog box.

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

VAL function

VAL (chars)

Converts a string to a number. The number's variable type is double. This function is the opposite of the **STR** function.

Parameter	Description
chars	The string expression to be converted. If the string does not begin with a number, VAL returns 0.

Note

- You can use only the period as a decimal separator with the VAL function. If you're not using a period (**.**) as a decimal separator, the **CDBL** or **CSNG** function can be used to convert a string to a number. Your Windows decimal settings are set in the Control Panel.
- The **VAL** function converts the string up to the first non-number character it encounters, from left to right in the string. Spaces are ignored.
- Because text box controls in dialog boxes can only return strings (even if numbers are input), you can use the **VAL** function to convert strings entered in dialog boxes to numbers.

Example

```
g = VAL("72nd Street")
h = VAL("72.700113")
```

Both the above statements assign 72 to the variables **g** and **h**.

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