Adobe® Photoshop® CS2

VBScript Scripting Reference

Adobe®

Adobe® Creative Suite 2 Photoshop® VBScript Scripting Reference for Windows® and Macintosh®.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Contents

1 Introduction

This reference describes the objects and commands in the Adobe® Photoshop® CS2 VBScript dictionary. A companion document, Photoshop CS2 Scripting Guide, describes basic scripting concepts and the Photoshop object model. This document provides reference details of the Photoshop object model, and additional information on VBScript-specific features.

This book contains the following sections:

- This introduction, which describes scripting support in Adobe Photoshop CS2, and lists changes to the VBScript interface since the previous release.
- VBScript Interface, which describes the objects of the VBScript type library for Adobe Photoshop CS2.
- Scripting Constants, which describes the enumerated values defined for use with Adobe Photoshop CS2 VBScript properties and methods.

Changes Since Earlier Versions

The following changes have been made to the VBScript object model and language support in Adobe Photoshop CS2:

- The emphasis has shifted from Visual Basic to VBScript with this version of Adobe Photoshop CS2.
- The following classes have been added to the VBScripting interface:
  - `CameraRawOpenOptions`, which you use to specify options when opening a document in Camera Raw format.
  - `ExportOptionsSaveForWeb`, which you use to optimize documents for the Web.
  - `ContactSheetOptions`, which you use to create and format contact sheets.
  - `BatchOptions`, which you use to specify options for the Batch command.
  - `LensBlurOptions`, which you use to specify options when applying the Lens Blur filter to a layer.
  - `Notifier` and `Notifiers`, which you use to associate a script with an event so that the script executes when the event occurs. For example, you can create a `Notifier` object to associate a script with the Photoshop CS2 application opening; whenever the application opens, the script runs.

2 VBScript Interface

The objects of the VBScript type library for Adobe Photoshop CS2 are presented alphabetically and in tabular format in this chapter.

Object properties and methods are described in separate tables for each object. See 'Working with the Properties Tables' on page 6 and 'Working with the Methods Tables' on page 6 for information on how to use these tables.

Sample code for several object model classes is given to help illustrate the syntax as well as usage of the object class.

Working with the Properties Tables

The Properties table for an object lists the following:

- The properties you can use with the object
- The value type for each property

  When the value type is a constant or another object, the value is a hypertext link to the constant's or object's listing, as in the following Properties table sample.
- The property's input status: Read-only or Read-write.
- A description that explains what the property is

  Descriptions are omitted for self-explanatory properties.

Property	Value Type	What it is
DisplayDialogs	PsDialogModes	Read-write. Controls whether or not Adobe Photoshop CS2 displays dialog boxes.

Working with the Methods Tables

The Methods table for an object lists the following:

- The method name
- Parameter(s)

  When a parameter type or return value is a constant or another object, the value is a hypertext link to the constant's or object's listing. In the following Methods table sample, the parameter type `ActionDescriptor` is an object; the parameter type `DialogModes` is a constant; the return value `ActionDescriptor` is also an object.

  Parameters can be required or optional. Optional parameters are indicated in the table by square brackets ([]). See 'Working with Method Parameters' on page 7 for information on using parameters.
- Return value type(s)
- A description, if applicable

Method	Parameter Type	Returns	What it does
ExecuteAction (EventID [, Descriptor] [, DisplayDialogs])	Number (Long) ActionDescriptor PsDialogModes	ActionDescriptor	Plays an ActionManager event.

Working with Method Parameters

Optional parameters are surrounded by square brackets ([ ]). In the following Methods table sample, the parameters Descriptor and DisplayDialogs are optional and the parameter EventID is not. See

Therefore, if you use the ExecuteAction() method for the object associated with the sample Methods table above, you *must* include an EventID value in the Parentheses following the method name. The EventID value must be a number, as indicated by the Number (Long) in the table's Parameter Type column.

If you use an optional parameter, you must separate the parameters with a comma, as indicated by the comma that precedes each optional parameter in the table.

Also, if you use an optional parameter, you must enter the values in the order they are listed in the table so that the JavaScript compiler knows which value you are entering. To skip an optional parameter, insert an extra comma to act as a placeholder.

The following sample provides values for an EventID and a DisplayDialog, but skips the Descriptor parameter (represented by the empty value between two commas). The statement executes action #4233 and allows only error type dialog boxes to be displayed.

```
appRef.ExecuteAction(4233,,Error)
```

ActionDescriptor

A record of key-value pairs for actions, such as those included on the Adobe Photoshop CS2 Actions menu.

Note: The `ActionDescriptor` class is part of the Action Manager functionality. See 'Action Manager' on page 145.

Properties

Property	Value Type	What it is
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Count	`Number (Long)`	Read-only. The number of keys contained in the descriptor.
typename	`String`	Read-only. The class name of the referenced `ActionDescriptor` object.

Methods

Method	Parameter Type	Returns	What it does
Clear ()			Clears the descriptor.
Erase (Key)	`Number (Long)`		Erases a key from the descriptor.
GetBoolean (Key)	`Number (Long)`	`Boolean`	Gets the value of a key of type boolean.
GetClass (Key)	`Number (Long)`	`Number (Long)`	Gets the value of a key of type class.
GetDouble (Key)	`Number (Long)`	`Number (Double)`	Gets the value of a key of type double.
GetEnumerationType (Key)	`Number (Long)`	`Number (Long)`	Gets the enumeration type of a key.
GetEnumerationValue (Key)	`Number (Long)`	`Number (Long)`	Gets the enumeration value of a key.
GetInteger (Key)	`Number (Long)`	`Number (Long)`	Gets the value of a key of type integer.
GetKey (Index)	`Number (Long)`	`Number (Long)`	Gets the ID of the *N*th key.
GetList (Key)	`Number (Long)`	`ActionList`	Gets the value of a key of type list.
GetObjectType (Key)	`Number (Long)`	`Number (Long)`	Gets the class ID of an object in a key of type object.

Method	Parameter Type	Returns	What it does (Continued)
GetObjectValue (Key)	Number (Long)	ActionDescriptor	Gets the value of a key of type object.
GetPath (Key)	Number (Long)	file	Gets the value of a key of type Alias.
GetReference (Key)	Number (Long)	ActionReference	Gets the value of a key of type ActionReference.
GetString (Key)	Number (Long)	String	Gets the value of a key of type String.
GetType (Key)	Number (Long)	PsDescValueType	Gets the type of a key.
GetUnitDoubleType (Key)	Number (Long)	Number (Long)	Gets the unit type of a key of type UnitDouble.
GetUnitDoubleValue (Key)	Number (Long)	Number (Double)	Gets the value of a key of type UnitDouble.
HasKey (Key)	Number (Long)	Boolean	Checks whether the descriptor contains the provided key.
IsEqual (otherDesc)	ActionDescriptor	Boolean	Determines whether the descriptor is the same as another descriptor.
PutBoolean (Key, Value)	Number (Long) Boolean		Sets the value for a key whose type is Boolean.
PutClass ((Key, Value)	Number (Long) Number (Long)		Sets the value for a key whose type is class.
PutDouble (Key, Value)	Number (Long) Number (Double)		Sets the value for a key whose type is double.
PutEnumerated (Key, EnumType, Value)	Number (Long) Number (Long) Number (Long)		Sets the enumeration type and value for a key. See Chapter 4, "Scripting Constants", for information on enumerated types.
PutInteger (Key, Value)	Number (Long) Number (Long)		Sets the value for a key whose type is integer.
PutList (Key, Value)	Number (Long) ActionList		Sets the value for a key whose type is an ActionList object.
PutObject (Key, ClassID, Value)	Number (Long) Number (Long) ActionDescriptor		Sets the value for a key whose type is an Action Descriptor.

Method	Parameter Type	Returns	What it does (Continued)
PutPath (Key, Value)	Number (Long) file		Sets the value for a key whose type is path.
PutReference (Key, Value)	Number (Long) ActionReference		Sets the value for a key whose type is an object reference.
PutString (Key, Value)	Number (Long) String		Sets the value for a key whose type is String.
PutUnitDouble (Key, UnitID, Value)	Number (Long) Number (Long) Number (Double)		Sets the value for a key whose type is a unit value formatted as a double.

ActionList

The list of commands that comprise an Action (such as an Action created using the Actions palette in the Adobe Photoshop CS2 application).

Note: The `actionList` object is part of the Action Manager functionality. For details on using the Action Manager, see 'Action Manager' on page 145.

Properties

Property	Value Type	What it is
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Count	`Number (Long)`	Read-only. The number of commands that comprise the action.
typename	`String`	Read-only. The class name of the referenced `actionList` object.

Methods

With the exception of the `Clear()` method, you use the methods of this object to either get the value of a specific type of data in the list or set (put) the value type.

Method	Parameter Type	Returns	What it does
Clear `()`			Clears the list.
GetBoolean `(Index)`	`Number (Long)`	`Boolean`	Gets the value of a list item of type boolean.
GetClass `(Index)`	`Number (Long)`	`Number (Long)`	Gets the value of a list item of type class.
GetDouble `(Index)`	`Number (Long)`	`Number (Double)`	Gets the value of a list item of type double.
GetEnumerationType `(Index)`	`Number (Long)`	`Number (Long)`	Gets the enumeration type of a list item.
GetEnumerationValue `(Index)`	`Number (Long)`	`Number (Long)`	Gets the enumeration value of a list item.
GetInteger `(Index)`	`Number (Long)`	`Number (Long)`	Gets the value of a list item of type integer.
GetList `(Index)`	`Number (Long)`	`ActionList`	Gets the value of a list item of type list.
GetObjectType `(Index)`	`Number (Long)`	`Number (Long)`	Gets the class ID of a list item of type object.

Method	Parameter Type	Returns	What it does (Continued)
GetObjectValue (Index)	Number (Long)	ActionDescriptor	Gets the value of a list item of type object.
GetPath (Index)	Number (Long)	file	Gets the value of a list item of type Alias.
GetReference (Index)	Number (Long)	ActionReference	Gets the value of a list item of type ActionReference.
GetString (Index)	Number (Long)	String	Gets the value of a list item of type String.
GetType (Index)	Number (Long)	PsDescValueType	Gets the type of a list item.
GetUnitDoubleType (Index)	Number (Long)	Number (Long)	Gets the unit value type of a list item of type Double.
GetUnitDoubleValue (Index)	Number (Long)	Number (Double)	Gets the unit value of a list item of type double.
PutBoolean (Value)	Boolean		Sets the value to either true or false.
PutClass (Value)	Number (Long)		Sets the class or data type.
PutDouble (Value)	Number (Double)		Sets the value type as a double.
PutEnumerated (EnumType, Value)	Number (Long) Number (Long)		Sets the value type as an enumerated, or constant, value. *constantType.VALUE* See Chapter 4, "Scripting Constants", for information on constant values.
PutInteger (Value)	Number (Long)		Sets the value of a list item of type integer.
PutList (Value)	ActionList		Sets the value of a list item of type list or array.
PutObject (ClassID, Value)	Number (Long) ActionDescriptor		Sets the value of a list item of type object.
PutPath (Value)	file		Sets the value of a list item of type path.
PutReference (Value)	ActionReference		Sets the value of a list item whose type a reference to an object created in the script.

Method	Parameter Type	Returns	What it does (Continued)
PutString (Value)	String		Sets the value of a list item of type String.
PutUnitDouble (UnitID, Value)	Number (Long) Number (Double)		Sets the value of a list item of type unit value represented as a double.

ActionReference

A reference object that contains the data describing the object you are referring to.

Note: The `actionReference` object is part of the Action Manager functionality. See 'Action Manager' on page 145.

Properties

Property	Value type	What it does
Application	Object (Application)	Read-only. The application that the collection belongs to.
typename	String	Read-only. The class name of the referenced Action object.

Methods

Method	Parameter Type	Returns	What it does
GetContainer ()		ActionReference	Gets the container object in the containment hierarchy for the object.
GetDesiredClass ()		Number (Long)	Gets a number representing the class of the object.
GetEnumeratedType ()		Number (Long)	Gets the enumeration type. See Chapter 4, "Scripting Constants", for information on enumeration types and values.
GetEnumeratedValue ()		Number (Long)	Gets the enumeration value.
GetForm ()		PsReferenceFormType	Gets the form of an ActionReference.
GetIdentifier ()		Number (Long)	Gets the identifier value for a reference whose form is identifier.
GetIndex ()		Number (Long)	Gets the index value for a reference in a list or array.
GetName ()		String	Gets the name of a reference.
GetOffset ()		Number (Long)	Gets the offset of the object's index value.
GetProperty ()		Number (Long)	Gets the property ID value.
PutClass (DesiredClass)	Number (Long)		Sets the class type of the object. The class name is required.

Method	Parameter Type	Returns	What it does (Continued)
PutEnumerated (DesiredClass, EnumType, Value)	Number (Long) Number (Long) Number (Long)		Sets the object's type to 'Enumerated'.
PutIdentifier (DesiredClass, Value)	Number (Long) Number (Long)		Sets the value of the identifier.
PutIndex (DesiredClass, Value)	Number (Long) Number (Long)		Sets the object's index value in a list.
PutName (DesiredClass, Value)	Number (Long) String		Sets the object's name.
PutOffset (DesiredClass, Value)	Number (Long) Number (Long)		Sets the object's offset from the current object.
PutProperty (DesiredClass, Value)	Number (Long) Number (Long)		Sets the value of the object's property.

Application

The Adobe Adobe Photoshop CS2 application object, referenced using the pre-defined global `app` object. The `app` object contains all other Adobe Photoshop CS2 objects.

Properties

Property	Value Type	What it is
ActiveDocument	Document	Read-write. The frontmost document. (Setting this property is equivalent to clicking an open document in the Adobe Photoshop CS2 application to bring it to the front of the screen.)
Application	Object (Application)	Read-only. The application that the collection belongs to.
BackgroundColor	SolidColor	Read-write. The color mode for the document's background color.
ColorSettings	String	Read-write. The name of selected color setting's set.
DisplayDialogs	PsDialogModes	Read-write. The dialog mode for the document, which indicates whether or not Adobe Photoshop CS2 displays dialogs when the script runs.
Documents	Documents	Read-only. The collection of open documents.
Fonts	TextFonts	Read-only. The fonts installed on this system.
ForegroundColor	SolidColor	Read-write. The default foreground color (used to paint, fill, and stroke selections).
FreeMemory	Number (Double)	Read-only. The amount of unused memory available to Adobe Photoshop CS2.
Locale	String	Read-only. The language location of the application.
MacintoshFileTypes	array of Strings	Read-only. A list of file image types Adobe Photoshop CS2 can open.
Name	String	Read-only. The application's name.
Notifiers	Notifiers	Read-only. The collection of notifiers currently configured (in the Scripts Events Manager menu in the Adobe Photoshop CS2 application).
NotifiersEnabled	Boolean	Read-write. Indication of whether all notifiers are enabled or disabled.
Path	file	Read-only. The full path to the location of the Adobe Photoshop CS2 application.
PlaybackDisplayDialogs	PsDialogModes	Read-write. The dialog mode for playback mode, which indicates whether or not Adobe Photoshop CS2 displays dialogs in playback mode.

Property	Value Type	What it is (Continued)
PlaybackParameters	ActionDescriptor	Read-write. The playback options, which indicate the speed at which Adobe Photoshop CS2 plays actions.
Preferences	Preferences	Read-only. The application preference settings (equivalent to selecting Edit > Preferences in the Adobe Photoshop CS2 application in Windows or Photoshop > Preferences in Mac OS).
PreferencesFolder	Alias	Read-only. The full path to the Preferences folder.
ScriptingVersion	String	Read-only. The version of the Scripting interface.
typename	String	Read-only. The class name of the referenced app object.
Version	String	Read-only. The version of Adobe Photoshop application you are running.
Visible	Boolean	Read-write. Indication of whether the Adobe Photoshop CS2 application is the front-most/active application.
WinColorSettings	String	Read-only. Color settings.
WindowsFileTypes	array of Strings	Read-only. A list of file image extensions Adobe Photoshop CS2 can open.

Methods

Method	Parameter Type	Returns	What it does
Batch (InputFiles, Action, From [, Options])	array of files String String BatchOptions	String	Runs the batch automation routine (similar to the Batch command, or File > Automate > Batch in the Adobe Photoshop CS2 application). **Note:** The inputFiles parameter specifies the source for the files to be manipulated by the Batch command.
ChangeColorSettings ([Name] [, File])	String Path to file name	None	Sets Color Settings to a named set or to the contents of a settings file.
CharIDToTypeID (CharID)	String	Number (Long)	Converts from a four character code (character ID) to a runtime ID.

Method	Parameter Type	Returns	What it does (Continued)
DoAction (Action, From)	String String	none	Plays an action from the Actions palette.
DoJavaScript (JavaScriptCode, [, Arguments] [, ExecutionMode]))	String String PsJavaScriptExecutionMode	String	Executes the specified JavaScript code.
DoJavaScriptFile (JavaScriptCode, [, Arguments] [, ExecutionMode]))	String String PsJavaScriptExecutionMode	String	Executes the specified JavaScript code.
executeAction (EventID [, Descriptor] [, DisplayDialogs])	Number (Long) ActionDescriptor PsDialogModes	ActionDescriptor	Plays an ActionManager event.
ExecuteActionGet (Reference)	ActionReference	ActionDescriptor	Obtains an ActionDescriptor.
Load (Document)	file		Loads a support document from the specified location.
MakeContactSheet (InputFiles [, Options])	array of files ContactSheetOptions	String	Creates a contact sheet from the specified files.
MakePDFPresentation (InputFiles OutputFiles [, Options])	array of files String PresentationOptions	String	Creates a PDF presentation file from the specified input files.
MakePhotoGallery (InputFolder OutputFolder [, Options])	String String GalleryOptions	String	Creates a Web photo gallery from the files in the specified input folder.
MakePhotomerge (InputFiles)	array of files	String	Merges multiple files into one; user interaction required.
MakePicturePackage (InputFiles [, Options])	array of files PicturePackageOptions	String	Creates a picture package from the specified input files.
Open (Document [, As])	String object (open options) **Note:** See open options for individual file types, such as CameraRawOpenOptions or EPSOpenOptions, etc.	Document	Opens the specified document as the optionally specified file type.

Method	Parameter Type	Returns	What it does (Continued)
Purge (Target)	PsPurgeTarget		Purges one or more caches.
StringIDToTypeID (StringID)	String	Number (Long)	Converts from a String ID to a runtime ID.
TypeIDToCharID (TypeID)	Number (Long)	String	Converts from a runtime ID to a character ID.
TypeIDToStringID (TypeID)	Number (Long)	String	Converts from a runtime ID to a String ID.

First Sample Script

The following script displays a message box that contains the application version number, the path to the application, the amount of memory available, and the number of documents open.

When the user clicks OK, a second dialog asks whether they would like the foreground and background colors set.

A third dialog offers to open a sample file. If the user clicks OK, the script opens the file eagle.psd from the samples folder in the application directory.

Application.vbs

```
' Create a Welcome message
' Use the name and version properties of the application object to append the
' application's name and version to the Welcome message use vbCrLf to insert a
' carriage return

Dim appRef, message, documentsOpen, answer, sampleDocToOpen

Set appRef = CreateObject("Photoshop.Application")

message = "Welcome to " & appRef.Name
message = message & " version " & appRef.Version & vbCrLf & vbCrLf

' find out where Photoshop CS2 is installed and add the path to the message add
' the optional parameter fsName to the path property to display the file system
' name in the most common format
message = message & "I'm installed in " & appRef.Path & vbCrLf & vbCrLf

'see how much memory Photoshop CS2 has to play with
message = message & "You have this much memory available for Photoshop CS2: "
message = message & appRef.FreeMemory & vbCrLf & vbCrLf

' use the Count property of the Documents object to see how many are open
documentsOpen = appRef.Documents.Count
message = message & "You currently have " & documentsOpen & " documents open." &
vbCrLf & vbCrLf

'display the message to the user
MsgBox (message)

answer = MsgBox ("Do you want me to set the foreground and background to my favorite
colors?", vbYesNo, "Change Colors?")
```

```
' set the colors
If answer = vbYes Then
   Randomize   ' Initialize random-number generator.
   ' I don't have a favorite color. Why did I ask you may wonder?
   appRef.ForegroundColor.RGB.Red = Rnd() * 255
   appRef.ForegroundColor.RGB.Green = Rnd() * 255
   appRef.ForegroundColor.RGB.Blue = Rnd() * 255
   appRef.BackgroundColor.RGB.Red = Rnd() * 255
   appRef.BackgroundColor.RGB.Green = Rnd() * 255
   appRef.BackgroundColor.RGB.Blue = Rnd() * 255
End If

' Open a document
If documentsOpen = 0 Then
   ' use the application's path and the offset to the samples folder
   sampleDocToOpen = appRef.Path & "Samples\Eagle.psd"
   ' compose a message with the name of the file
   message = "Would you like me to open a sample for you? ("
   message = message & sampleDocToOpen & ")"
   ' ask the user another question
   answer = MsgBox (message, vbYesNo, "Open Something?")
   ' open the document accordingly
   If answer = vbYes Then
      appRef.Open sampleDocToOpen
   End If
End If
```

Second Sample Script

The following script presents a progression of images as a PDF slide show.

PDFPresentation.vbs

```
' use all the files in the Samples folder
Dim appRef, inputFiles(), i, outputFile, options

Set appRef = CreateObject("Photoshop.Application")

' get all the files found in this folder
Set fsoRef = CreateObject( "Scripting.FileSystemObject" )
Set folderRef = fsoRef.GetFolder( appRef.Path & "Samples\" )

ReDim inputFiles(folderRef.Files.Count)

i = 0
For Each f in folderRef.Files
   inputFiles(i) = f.Path
   i = i + 1
Next

' output to the desktop
outputFile = "C:\\JavaScriptPresentation.pdf"

' there are defaults but I like to set the options myself
Set options = CreateObject("Photoshop.PresentationOptions")
options.Presentation = true
options.Encoding = 2 'for PsPDFEncoding --> 2 (psPDFJPEG)
options.AutoAdvance = true
options.Interval = 5
options.Loop = true
options.Transition = 10 'for PsTransitionType --> 10 (psRandom)

' create the presentation
appRef.MakePDFPresentation inputFiles, outputFile, options
```

ArtLayer

An object within a document that contains the visual elements of the image (equivalent to a layer in the Adobe Photoshop CS2 application).

Properties

Property	Value Type	What it is
AllLocked	Boolean	Read-write. Indicates whether to completely lock the layer's contents and settings.
Application	Object (Application)	Read-only. The application that the collection belongs to.
ArtLayer	Object (ArtLayer)	Read-only. When `LayerType = 1`, then this property returns a reference to the corresponding ArtLayer object. See LayerType.
BlendMode	PsBlendMode	Read-write. The layer's blending mode.
Bounds	Array	Read-only. An array of coordinates that describes the bounding rectangle of the ArtLayer.
FillOpacity	Number (Double)	Read-write. The interior opacity of the layer (0.0 - 100.0).
Grouped	Boolean	Read-write. Indication of whether to group this layer with the layer beneath it.
IsBackgroundLayer	Boolean	Read-write. Indicates whether the layer is a background layer or normal layer. **Note:** A document can have only one background layer.
Kind	PsLayerKind	Read-write. Sets the layer's kind (such as 'text layer') for an empty layer. **Note:** Valid only when the layer is empty and when `IsBackgroundLayer` is `false`. See IsBackgroundLayer. **Note:** You can use the `kind` property to make a background layer a normal layer; however, to make a layer a background layer, you must set `IsBackgroundLayer` to `true`.
Layer	Object (Layer)	Read-only. The Layer corresponding to the ArtLayer.
LayerSet	Object (LayerSet)	Read-only. When `LayerType = 2`, then this property returns a reference to the corresponding LayerSet object. See LayerType.
LayerType	PsLayerType	Read-write. The type of layer.

Property	Value Type	What it is (Continued)
LinkedLayers	array of Layers	Read-only. The layers linked to this layer. **Note:** See Link.
Name	String	Read-write. The layer's name.
Opacity	Number (Double)	Read-write. The master opacity of the layer (0.0 - 100.0).
Parent	Object (Document)	Read-only. The object's container.
PixelsLocked	Boolean	Read-write. Indicates whether the pixels in the layer's image can be edited using the paintbrush tool.
PositionLocked	Boolean	Read-write. Indicates whether the pixels in the layer's image can be moved within the layer.
TextItem	TextItem	Read-only. The text item that is associated with the layer. **Note:** Valid only when Kind = 2. See Kind.
TransparentPixelsLocked	Boolean	Read-write. Indicates whether editing is confined to the opaque portions of the layer.
typename	String	Read-only. The class name of the referenced ArtLayer object.
Visible	Boolean	Read-write. Indicates whether the layer is visible.

Methods

Method	Parameter Type	Returns	What it does
AdjustBrightnessContrast (Brightness, Contrast)	Number (Long) Number (Long)		Adjusts the brightness (-100 - 100) and contrast (-100 - 100).
AdjustColorBalance ([Shadows] [, Midtones] [, Highlights] [, PreserveLuminosity]	array of Numbers array of Numbers array of Numbers Boolean		Adjusts the color balance of the layer's component channels. For Shadows, Midtones, and Highlights, the array must include three values (-100 - 100), which represent cyan or red, magenta or green, and yellow or blue, when the document mode is CMYK or RGB. **Note:** See mode in the Properties table of the Document object.

Method	Parameter Type	Returns	What it does (Continued)
AdjustCurves (CurveShape)	Array of points (Array (Array(x, y)))		Adjusts the tonal range of the selected channel using up to fourteen points.
AdjustLevels (InputRangeStart, InputRangeEnd, InputRangeGamma, OutputRangeStart, OutputRangeEnd)	Number (Long) Number (Long) Number (Double) Number (Long) Number (Long)		Adjusts the levels of the selected channels (InputRangeStart: 0 - 253; InputRangeEnd: (InputRangeStart + 2) - 255; InputRangeGamma: 0.10 - 9.99; OutputRangeStart: 0 - 253; OutputRangeEnd: (OutputRangeStart + 2) - 255.
ApplyAddNoise (Amount, Distribution, Monochromatic)	Number (Double) PsNoiseDistribution Boolean		Applies the Add Noise filter (Amount: 0.1% - 400%).
ApplyAverage ()			Applies the Average filter.
ApplyBlur ()			Applies the Blur filter.
ApplyBlurMore ()			Applies the Blur More filter.
ApplyClouds ()			Applies the Clouds filter.
ApplyCustomFilter (Characteristics, Scale, Offset)	Array of twenty-five Numbers (Long) Number (Long) Number (Long)		Applies a custom filter. **Note:** Required parameter values define the filter. Refer to Adobe Photoshop CS2 Help for specific instructions.
ApplyDeInterlace (EliminateFields, CreateFields)	PsEliminateFields PsCreateFields		Applies the De-Interlace filter.
ApplyDespeckle ()			Applies the Despeckle filter.
ApplyDifferenceClouds ()			Applies the Difference Clouds filter.
ApplyDiffuseGlow (Graininess, GlowAmount, ClearAmount)	Number (Long) Number (Long) Number (Long)		Applies the Diffuse Glow filter (Graininess: 0 - 10; GlowAmount: 0 - 20; ClearAmount: 0 - 20).

Method	Parameter Type	Returns	What it does (Continued)
ApplyDisplace (AmountScale, VerticalScale, DisplacementType, UndefinedAreas, DisplacementMapFiles)	Number (Long) Number (Long) PsDisplacementMapType PsUndefinedAreas String		Applies the Displace filter using the specified horizontal and vertical scale (-999 - 999), mapping type, treatment of undistorted areas, and path to the distortion image map.
ApplyDustAndScratches (Radius, Threshold)	Number (Long) Number (Long)		Applies the Dust & Scratches filter (Radius: 1 - 100; Threshold: 0 - 255).
ApplyGaussianBlur (Radius)	Number (Double)		Applies the Gaussian Blur filter within the specified radius (in pixels) (0.1 - 250.0).
ApplyGlassEffect (Distortion, Smoothness, Scaling [, Invert] [, Texture] [, TextureFile])	Number (Long) Number (Long) Number (Long) Boolean PsTextureType File		Applies the Glass filter (Distortion: 0 - 20; Smoothness: 1 - 15; Scaling (in percent): 50 - 200).
ApplyHighPass (Radius)	Number (Double)		Applies the High Pass filter within the specified radius (in pixels) (0.1 - 250.0).
ApplyLensBlur ([Options])	LensBlurOptions		Applies the Lens Blur filter.
ApplyLensFlare (Brightness, FlareCenter, LensType)	Number (Long) Array PsLensType		Applies the Lens Flare filter with the specified brightness (0 - 300%), the x and y coordinates (unit value) of the flare center, and the lens type.
ApplyMaximum (Radius)	Number (Double)		Applies the Maximum filter within the specified radius (in pixels) (1 - 100).
ApplyMedianNoise (Radius)	Number (Double)		Applies the Median Noise filter within the specified radius (in pixels) (1 - 100).
ApplyMinimum (Radius)	Number (Double)		Applies the Minimum filter within the specified radius (in pixels) (1 - 100).
ApplyMotionBlur (Angle, Radius)	Number (Long) Number (Double)		Applies the Motion Blur filter (Angle: -360 - 360; Radius: 1 - 999).
ApplyNTSC ()			Applies the NTSC colors filter.

Method	Parameter Type	Returns	What it does (Continued)
ApplyOceanRipple (Size, Magnitude)	Number (Long) Number (Long)		Applies the Ocean Ripple filter in the specified size (1 - 15) and magnitude (0 - 20).
ApplyOffset (Horizontal, Vertical, UndefinedAreas)	Number (Double) Number (Double) PsOffsetUndefinedAreas		Moves the layer the specified amount horizontally and vertically (min/max amounts depend on layer size), leaving an undefined area at the layer's original location.
ApplyPinch (Amount)	Number (Long)		Applies the Pinch filter in the specified amount (as a percentage) (-100 - 100).
ApplyPolarCoordinates (Conversion)	PsPolarConversionType		Applies the Polar Coordinates filter.
ApplyRadialBlur (Amount, BlurMethod, BlurQuality)	Number (Long) PsRadialBlurMethod PsRadialBlurQuality		Applies the Radial Blur filter in the specified amount (1 - 100) using either a spin or zoom effect and the specified quality.
ApplyRipple (Amount, Size)	Number (Long) PsRippleSize		Applies the Ripple filter in the specified amount (-999 to 999) throughout the image and in the specified size.
ApplySharpen ()			Applies the Sharpen filter.
ApplySharpenEdges ()			Applies the Sharpen Edges filter.
ApplySharpenMore ()			Applies the Sharpen More filter.
ApplyShear (Curve, UndefinedAreas)	array of points (Array (Array(x, y))) PsUndefinedAreas		Applies the Shear filter (curve: 2 - 255 points). **Note:** You must define at least two points in the Curve parameter.
ApplySmartBlur (Radius, Threshold, BlurQuality, Mode)	Number (Double) Number (Double) PsSmartBlurQuality PsSmartBlurMode		Applies the smart blur filter (Radius: 0.1 - 100.0; Threshold: 0.1 - 100.0).
ApplySpherize (Amount, Mode)	Number (Long) PsSpherizeMode		Applies the Spherize filter in the specified amount (as percentage) (-100 - 100).

Method	Parameter Type	Returns	What it does (Continued)
ApplyStyle (StyleName)	String		Applies the specified style to the layer. **Note:** You must use a style from the Styles list in the Layer Style dialog.
ApplyTextureFill (TextureFile)	String		Applies the Texture Fill filter.
ApplyTwirl (Angle)	Number (Long)		Applies the Twirl filter at the specified angle (-999 - 999).
ApplyUnSharpMask (Amount, Radius, Threshold)	Number (Double) Number (Double) Number (Long)		Applies the Unsharp Mask filter (Amount: 1 - 500 as percent; Radius: 0.1 - 250.00; Threshold: 0 - 255).
ApplyWave (GeneratorNumber, MinimumWavelength, MaximumWavelength, MinimumAmplitude, MaximumAmplitude, HorizontalScale, VerticalScale, WaveType, UndefinedAreas, RandomSeed)	Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) PsWaveType PsUndefinedAreas Number (Long)		Applies the Wave filter (GeneratorNumber**: 1 - 999;** MinimumWavelength: **1 - 998;** MaximumWavelength**:** **2 -** MinimumWavelength + 1**;** MinimumAmplitude**: 1 - 998;** MaximumAmplitude**:** **2 -** MinimumAmplitude + 1**;** AmountScale**: 1% - 100%;** VerticalScale**: 1% - 100%**).
ApplyZigZag (Amount, Ridges, Style)	Number (Long) Number (Long) PsZigZagType		Applies the Zigzag filter (Amount: -100 - 100; Ridges: 0 - 20).
AutoContrast ()			Adjusts the contrast of the selected channels automatically.
AutoLevels ()			Adjusts the levels of the selected channels using the auto levels option.
Clear ()			Cuts the layer without moving it to the clipboard.
Copy ([Merge])	Boolean		Copies the layer to the clipboard. When the optional argument is set to true, a merged copy is performed (that is, all visible layers are copied to the clipboard).
Cut ()			Cuts the layer to the clipboard.

Method	Parameter Type	Returns	What it does (Continued)
Desaturate ()			Converts a color image to a grayscale image in the current color mode by assigning equal values of each component color to each pixel.
Duplicate ([RelativeObject] [, InsertionLocation])	Object (Layer) PsElementPlacement	Object (Layer)	Creates a duplicate of the object on the screen.
Equalize ()			Redistributes the brightness values of pixels in an image to more evenly represent the entire range of brightness levels within the image.
Invert ()			Inverts the colors in the layer by converting the brightness value of each pixel in the channels to the inverse value on the 256-step color-values scale.
Link ()	Object (Layer)		Links the layer with the specified layer.
Merge ()		ArtLayer	Merges the layer down, removing the layer from the document; returns a reference to the art layer that this layer is merged into.
MixChannels (OutputChannels [, Monochrome])	array of array of numbers (double) boolean		Modifies a targeted (output) color channel using a mix of the existing color channels in the image. (OutputChannels = An array of channel specifications. For each component channel, specify a list of adjustment values (-200 - 200) followed by a 'constant' value (-200 - 200).) **Note:** When Monochrome = true, the maximum number of channel value specifications is 1. **Note:** Valid only when Document.Mode = 2 or Document.Mode = 3. **Note:** RGB arrays must include four doubles. CMYK arrays must include five doubles.

Method	Parameter Type	Returns	What it does (Continued)
Move (ApplicationObject, InsertionLocation)	Object (ArtLayer or LayerSet) PsElementPlacement		Moves the layer relative to the object specified in parameters. **Note:** For art layers, only the constant values 3 and 2 are valid. For layer sets, only the constant values 3 and 0 are valid.
PhotoFilter ([FillColor] [, Density] [, PreserveLuminosity])	SolidColor Number (Long) Boolean		Adjust the layer's color balance and temperature as if a color filter had been applied (Density: 1% - 100%).
Posterize (Levels)	Number (Long)		Specifies the number of tonal levels (2 - 255) for each channel and then maps pixels to the closest matching level.
Rasterize (Target)	PsPurgeTarget		Converts the targeted contents in the layer into a flat, raster image.
Resize ([Horizontal] [, Vertical] [, Anchor])	Number (Double) Number (Double) PsAnchorPosition		Resizes the layer to the specified dimensions (as a percentage of its current size) and places it in the specified position.
Rotate (Angle [, Anchor])	Number (Double) PsAnchorPosition		Rotates rotates the layer around the specified anchor point.
SelectiveColor (SelectionMethod [, Reds] [, Yellows] [, Greens] [, Cyans] [, Blues] [, Magentas] [, Whites] [, Neutrals] [, Blacks])	PsAdjustmentReference Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long) Array of Numbers (Long)		Modifies the amount of a process color in a specified primary color without affecting the other primary colors. **Note:** Each color array must have four components.

Method	Parameter Type	Returns	What it does (Continued)
ShadowHighlight ([ShadowAmount] [, ShadowWidth] [, ShadowRadius] [, HighlightAmount] [, HighlightWidth] [, HighlightRadius] [, ColorCorrection] [, MidtoneContrast] [, BlackClip] [, WhiteClip])	Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) Number (Long) Number (Double) Number (Double)		Adjusts the range of tones in the image's Shadows and highlights (ShadowAmount: 0 - 100 as percent; ShadowWidth: 0 - 100 as percent; ShadowRadius: 0 - 2500 in pixels; HighlightAmount: 0 - 100 as percent; HighlightWidth: 0 - 100 as percent; HighlightRadius: 0 - 2500 in pixels; ColorCorrection: -100 - 100; MidtoneContrast: -100 - 100; BlackClip: 0.000 - 50.000; WhiteClip: 0.000 - 50.000).
Threshold (Level)	Number (Long)		Converts grayscale or color images to high-contrast, B/W images by converting pixels lighter than the specified threshold to white and pixels darker than the threshold to black (level: 1 - 255).
translate ([DeltaX] [, DeltaY])	UnitValue UnitValue		Moves the layer the specified amount (in pixels) relative to its current position.
unlink ()			Unlinks the layer.

Sample Script

The following script closes any open documents (files), then opens each file in the Samples folder, creating an art layer for each image and using the image's file name as the layer name.

ArtLayer.vbs

```
Dim appRef, startRulerUnits, startTypeUnits, startDisplayDialogs, mergedDoc
Dim fsoRef, folderRef, topLeftH, topLeftV, docH, docV, docName, selRegion
Dim x, y, layer1, layer2

Set appRef = CreateObject("Photoshop.Application")

' Save the current preferences
startRulerUnits = appRef.Preferences.RulerUnits
startTypeUnits = appRef.Preferences.TypeUnits
startDisplayDialogs = appRef.DisplayDialogs

' Set Photoshop CS2 to use pixels and display no dialogs
appRef.Preferences.RulerUnits = 1 'for PsUnits --> 1 (psPixels)
appRef.Preferences.TypeUnits = 1 'for PsTypeUnits --> 1 (psPixels)
appRef.DisplayDialogs = 3 'for PsDialogModes --> 3 (psDisplayNoDialogs)

' Close all the open documents
Do While appRef.Documents.Count
   appRef.ActiveDocument.Close()
Loop
```

```
' Create a new document to merge all the samples into
Set mergedDoc = appRef.Documents.Add(1000, 1000, 72, "Merged Samples", 2, 3, 1)
'enumerated values 2 = PsNewDocumentMode --> 2 (PsNewRGB) and
'3 = PsDocumentFill --> 3 (psTransparent)

' get all the files found in this folder
Set fsoRef = CreateObject( "Scripting.FileSystemObject" )
Set folderRef = fsoRef.GetFolder( appRef.Path & "Samples\" )

Randomize

' open each file
For Each f in folderRef.Files
   appRef.Open f.Path

   ' use the document name for the layer name in the merged document
   docName = appRef.ActiveDocument.Name

   ' flatten the document so we get everything and then copy
   appRef.ActiveDocument.flatten()
   appRef.ActiveDocument.Selection.SelectAll()
   appRef.ActiveDocument.Selection.Copy()

   ' don't save anything we did
   appRef.ActiveDocument.Close(2)
   'the enumerated value Close(2) = PsSaveOptions >2 (psDoNotSaveChanges)

   ' make a random selection on the document to paste into
   ' by dividing the document up in 4 quadrants and pasting
   ' into one of them by selecting that area
   topLeftH = Int(Rnd() * 2)
   topLeftV = Int(Rnd() * 2)
   ' MsgBox topLeftH & ":" & topLeftV
   docH = appRef.ActiveDocument.Width / 2
   docV = appRef.ActiveDocument.Height / 2
   selRegion = Array( Array( topLeftH * docH, topLeftV * docV), _
                  Array( topLeftH * docH + docH, topLeftV * docV), _
                  Array( topLeftH * docH + docH, topLeftV * docV + docV), _
                  Array( topLeftH * docH, topLeftV * docV + docV), _
                  Array( topLeftH * docH, topLeftV * docV))
   appRef.ActiveDocument.Selection.Select(selRegion)
   appRef.ActiveDocument.Paste()

   ' change the layer name and opacity
   appRef.ActiveDocument.ActiveLayer.Name = docName
   appRef.ActiveDocument.ActiveLayer.FillOpacity = 50
Next

' sort the layers by name
x = 0
y = 0
for x = 1 To appRef.ActiveDocument.Layers.Count
   for y = 1 To appRef.ActiveDocument.Layers.Count - 1
      Set layer1 = appRef.ActiveDocument.Layers(y)
      Set layer2 = appRef.ActiveDocument.Layers(y + 1)
      If layer1.Name --> layer2.Name Then
         layer1.move layer2, 4
      End If
   Next
```

```
Next

' Reset the application preferences
appRef.Preferences.RulerUnits = startRulerUnits
appRef.Preferences.TypeUnits = startTypeUnits
appRef.DisplayDialogs = startDisplayDialogs
```

ArtLayers

The collection of `ArtLayer` objects in the document.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `ArtLayers` collection.
Item	Object (ArtLayer)	Read-only. Used to get the element from the collection.
Parent	Object (Document)	Read-only. The object's container.
typename	String	Read-only. The class name of the referenced `ArtLayers` object.

Methods

Method	Parameter Type	Returns	What it does
Add ()		ArtLayer	Creates a new `ArtLayer` in the document.
Index (ItemPtr)	Object (ArtLayer)	Number (long)	Gets an element from the `ArtLayers` collection.
RemoveAll ()		Nothing	Removes all elements from the `ArtLayers` collection.

BatchOptions

Options to specify when running a Batch command.

Properties

Property	Value type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Destination	PsBatchDestinationType	Read-write. The type of destination for the processed files. Default: 1 (psNoDestination).
DestinationFolder	file	Read-write. The folder location for the processed files. **Note:** Valid only when Destination = 3 (psFolder). See Destination.
ErrorFile	file	Read-write. The file in which to log errors encountered. **Note:** To display errors on the screen (and stop batch processing when errors occur) leave blank.
FileNaming	Array of (PsFileNamingType options)	Read-write. A list of file naming options (maximum: 6). **Note:** Valid only when Destination = 3 (psFolder). See Destination.
MacintoshCompatible	Boolean	Read-write. Indication of whether to make the final file names Macintosh compatible. Default: true. **Note:** Valid only when Destination = 3 (psFolder). See Destination.
OverrideOpen	Boolean	Read-write. Indication of whether to override action open commands. Default: false.
OverrideSave	Boolean	Read-write. Indication of whether to override save as action steps with the specified destination. Default: false. **Note:** Valid only when Destination = 3 (psFolder). or Destination = 2 (psSaveAndClose). See Destination.
StartingSerial	Number (Long)	Read-write. The starting serial number to use in naming files. Default: 1. **Note:** Valid only when Destination = 3 (psFolder). See Destination.

Property	Value type	What it is (Continued)
SuppressOpen	Boolean	Read-write. Indication of whether to suppress the file open options dialogs. Default: false.
SuppressProfile	Boolean	Read-write. Indication of whether to suppress the color profile warnings. Default: false.
typename	String	Read-only. The class name of the referenced batchOptions object.
UnixCompatible	Boolean	Read-write. Indication of whether to make the final file name Unix® compatible. Default: true. **Note:** Valid only when Destination = 3 (psFolder). See Destination.
WindowsCompatible	Boolean	Read-write. Indication of whether to make the final file names Windows compatible. Default: true. **Note:** Valid only when Destination = 3 (psFolder). See Destination.

BitmapConversionOptions

Options to be specified when converting an image to Bitmap mode.

Note: Convert color images to grayscale before converting the image to bitmap mode. See 'desaturate' on page 62 (in the Properties table of the `ArtLayer` object).

Properties

Property	Value Type	What it is
Angle	`Number (Double)`	Read-write. The angle (in degrees) at which to orient individual dots (-180 - 180). See Shape. **Note:** Valid only when `Method = 4`. See Method.
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Frequency	`Number (Double)`	Read-write. The number of printer dots (per inch) to use (1.0 - 999.99). **Note:** Valid only when `Method = 4`. See Method.
Method	`PsBitmapConversionType`	Read-write. The conversion method to use. Default: `3`.
PatternName	`String`	Read-write. The name of the pattern to use. **Note:** Valid only when `Method = 5`. See Method.
Resolution	`Number (Double)`	Read-write. The output resolution in pixels per inch. Default: `72.0`.
Shape	`PsBitmapHalfToneType`	Read-write. The dot shape to use. **Note:** Valid only when `Method = 1`. See Method.
typename	`String`	Read-only. The class name of the referenced `BitmapConversionOptions` object.

BMPSaveOptions

Options that can be specified when saving a document in BMP format.

Properties

Property	Value Type	What it is
AlphaChannelsd	Boolean	Read-write. Indication of whether to save the alpha channels.
Application	Object (Application)	Read-only. The application that the collection belongs to.
Depth	PsBitmapConversionType	Read-write. The number of bits per channel.
FlipRowOrder	Boolean	Read-write. Indication of whether to write the image from top to bottom. Default: false. **Note:** Available only when OSType = 2. See OSType.
OSType	PsOperatingSystem	Read-write. The target OS. Default: 2.
RLECompression	Boolean	Read-write. Indication of whether to use RLE compression. **Note:** Available only when OSType = 2. See OSType.
typename	String	Read-only. The class name of the referenced BMPSaveOptions object.

CameraRawOpenOptions

Options that can be specified when opening a document in Camera Raw format.

Properties

Property	Value type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
BitsPerChannel	PsBitsPerChannelType	Read-write. The number of bits per channel.
BlueHue	Number (Long)	Read-write. The blue hue of the shot (-100 - 100).
BlueSaturation	Number (Long)	Read-write. The blue saturation of the shot (-100 - 100).
Brightness	Number (Long)	Read-write. The brightness of the shot (0 - 150).
ChromaticAberrationBY	Number (Long)	Read-write. The chromatic aberration B/Y of the shot (-100 - 100).
ChromaticAberrationRC	Number (Long)	Read-write. The chromatic aberration R/C of the shot (-100 - 100).
ColorNoiseReduction	Number (Long)	Read-write. The color noise reduction of the shot (0 - 100).
ColorSpace	PsColorSpaceType	Read-write. The colorspace for the image.
Contrast	Number (Long)	Read-write. The contrast of the shot (-50 - 100).
Exposure	Number (Double)	Read-write. The exposure of the shot (4.0 - 4.0).
GreenHue	Number (Long)	Read-write. The green hue of the shot (-100 - 100).
GreenSaturation	Number (Long)	Read-write. The green saturation of the shot (-100 - 100).
LuminanceSmoothing	Number (Long)	Read-write. The luminance smoothing of the shot (0 - 100).
RedHue	Number (Long)	Read-write. The red hue of the shot (-100 - 100).
RedSaturation	Number (Long)	Read-write. The red saturation of the shot (-100 - 100).
Resolution	Number (Double)	Read-write. The resolution of the document in pixels per inch (1 - 999).
Saturation	Number (Long)	Read-write. The saturation of the shot (-100 - 100).
Settings	PsCameraRAWSettingsType	Read-write. The global settings for all Camera RAW options.
Shadows	Number (Long)	Read-write. The shadows of the shot (0 - 100).

Property	Value type	What it is (Continued)
ShadowTint	Number (Long)	Read-write. The shadow tint of the shot (-100 - 100).
Sharpness	Number (Long)	Read-write. The sharpness of the shot (0 - 100).
Size	PsCameraRAWSize	Read-write. The size of the new document.
Temperature	Number (Long)	Read-write. The temperature of the shot (2000 - 50000).
Tint	Number (Long)	Read-write. The tint of the shot (-150 - 150).
typename	String	Read-only. The class name of the referenced CameraRawOpenOptions object.
VignettingAmount	Number (Long)	Read-write. The vignetting amount of the shot (-100 - 100).
VignettingMidpoint	Number (Long)	Read-write. The vignetting mid point of the shot (-100 - 100).
WhiteBalance	PsWhiteBalanceType	Read-write. The white balance options for the image.

Channel

Object that stores information about a color element in the image, analogous to a plate in the printing process that applies a single color. The document's color mode determines the number of default channels; for example, an RGB document has four default channels:

- A composite channel: RGB
- Three component channels: red, green, blue

A channel can also be an alpha channel, which stores selections as masks, or a spot channel, which stores spot colors.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Color	Object (SolidColor)	Read-write. The color of the channel. **Note:** Not valid when Type = 1.
Histogram	Array of 256 Numbers (Long)	Read-only. A histogram of the color of the channel. **Note:** Not valid when Type = 1. For component channel histogram values, use the Histogram property of the Document object instead. See Histogram.
Kind	PsChannelType	Read-write. The channel type.
Name	String	Read-write. The channel's name.
Opacity	Number (Double)	Read-write. The opacity to use for alpha channels or the solidity to use for spot channels (0 - 100). **Note:** Valid only when Type = 2 or Type = 3.
Parent	Object (Document)	Read-only. The object's container.
typename	String	Read-only. The class name of the referenced Channel object.
Visible	Boolean	Read-write. Indicates whether the channel is visible.

Methods

Method	Parameter Type	Returns	What it does
Delete ()			Deletes the channel.
Duplicate ([TargetDocument])	Document	Channel	Duplicates the channel.
Merge ()			Merges a spot channel into the component channels.

Channels

The collection of `Channel` objects in the document. See 'Channel' on page 40.

Properties

Property	Value Type	What it is
`Application`	`Object (Application)`	Read-only. The application that the collection belongs to.
`Count`	`Number (Long)`	Read-only. The number of elements in the `Channels` collection.
`Item`	`Object (Channel)`	Read-only. Gets an element from the collection.
`typename`	`String`	Read-only. The class name of the referenced `Channels` object.

Methods

Method	Parameter Type	Returns	What it does
`Add` `()`		`Channel`	Creates a new `Channel` object.
`Index` `(ItemPtr)`	`Object (Channel)`	`Number (Long)`	
`RemoveAll` `()`			Removes all `Channel` objects from the `Channels` collection.

Sample Script

The following script checks for any open documents (files); if no documents are open, it opens `eagle.psd` in the Samples folder.

The script then creates the histogram for each channel in the document and creates a log file named `Histogram.log` on your C:\ drive. The file maps out the histogram for each channel and includes the following information:

- Total pixel count
- Mean pixel count per row
- Standard deviation of pixels per row
- Median pixels per row
- Pixel count row by row, represented in rows of Xs, produces a strobe effect, as a progression of dialogs display.

Histogram.vbs

```
Dim appRef, startRulerUnits, startTypeUnits, startDisplayDialogs, docRef
Dim totalCount, channelIndex, activeChannels, myChannels, secondaryIndex
Dim largestCount, histogramIndex, pixelsPerX, outputX, a, visibleChannelCount
```

```
Dim aChannelArray(), aChannelIndex, oFileSys, fileOut, hist

Set appRef = CreateObject("Photoshop.Application")

' Save the current preferences
startRulerUnits = appRef.Preferences.RulerUnits
startTypeUnits = appRef.Preferences.TypeUnits
startDisplayDialogs = appRef.DisplayDialogs

' Set Photoshop CS2 to use pixels and display no dialogs
appRef.Preferences.RulerUnits = 1 'for PsUnits --> 1 (psPixels)
appRef.Preferences.TypeUnits = 1 'for PsTypeUnits --> 1 (psPixels)
appRef.DisplayDialogs = 3 'for PsDialogModes --> 3 (psDisplayNoDialogs)

' if there are no documents open then try to open a sample file
If appRef.Documents.Count = 0 Then
   appRef.Open(appRef.Path + "/Samples/Eagle.psd")
End If

Set docRef = appRef.ActiveDocument

' create the output file
Set oFileSys = CreateObject("Scripting.FileSystemObject")
Set fileOut = oFileSys.CreateTextFile("C:\\Histogram.log")

' write out a header
fileOut.Write "Histogram report for " & docRef.Name

' find out how many pixels I have
totalCount = docRef.Width * docRef.Height

' more info to the out file
fileOut.WriteLine " with a total pixel count of " & totalCount

' remember which channels are currently active
activeChannels = appRef.ActiveDocument.ActiveChannels

' document histogram only works in these modes
If docRef.Mode = 2 Or docRef.Mode = 3 Or docRef.Mode = 6 Then
'enumerated values = PsDocumentMode --> 2 (psRGB), 3 (psCMYK), 6 (psIndexedColor)

   ' activate the main channels so we can get the document's histogram
   ' using the TurnOnDocumentHistogramChannels function
   Call TurnOnDocumentHistogramChannels(docRef)

   ' Output the documents histogram
   Call OutputHistogram(docRef.Histogram, "Luminosity", fileOut)
End If

' local reference to work from
Set myChannels = docRef.Channels

' loop through each channel and output the histogram
For channelIndex = 1 To myChannels.Count
   ' the channel has to be visible to get a histogram
   myChannels(channelIndex).Visible = true

   ' turn off all the other channels
   for secondaryIndex = 1 to myChannels.Count
       If Not channelIndex = secondaryIndex Then
```

```
            myChannels(secondaryIndex).Visible = false
        End If
    Next

    ' Use the function to dump the histogram
    Call OutputHistogram(myChannels(channelIndex).Histogram,
myChannels(channelIndex).Name, fileOut)
Next

' close down the output file
fileOut.Close

' reset the active channels
docRef.ActiveChannels = activeChannels

' Reset the application preferences
appRef.Preferences.RulerUnits = startRulerUnits
appRef.Preferences.TypeUnits = startTypeUnits
appRef.DisplayDialogs = startDisplayDialogs


' Utility function that takes a histogram and name
' and dumps to the output file
Private Function OutputHistogram (inHistogram, inHistogramName, inOutFile)
    ' find out which count has the largest number
    ' I scale everything to this number for the output
    largestCount = 0

    ' a simple indexer I can reuse
    histogramIndex = 0

    ' search through all and find the largest single item
    For Each hist In inHistogram
        histogramCount = histogramCount + CLng(hist)
        If CLng(hist) --> largestCount Then
            largestCount = CLng(hist)
        End If
    Next

    'These should match
    If Not histogramCount = totalCount Then
        MsgBox "Something bad is happening!"
    End If

    inOutFile.WriteLine "This histogram has a pixel count of " & histogramCount
    inOutFile.WriteLine

    'see how much each "X" is going to count as
    pixelsPerX = largestCount / 100

    'output this data to the file
    inOutFile.WriteLine "One X = " & pixelsPerX & " pixels."

    'output the name of this histogram
    inOutFile.WriteLine inHistogramName

    inOutFile.WriteLine "Mean Pixels: " & AverageHistogram(inHistogram)
    inOutFile.WriteLine "Std. Dev. Pixels: " &
StandardDeviationHistogram(inHistogram)
```

```
   inOutFile.WriteLine "Median Pixels: " & MedianHistogram(inHistogram,
histogramCount)

   ' loop through all the items and output in the following format
   ' 001
   ' 002
   ' For histogramIndex = 0 To (inHistogram.Count - 1)
   histogramIndex = 0
   For Each hist in inHistogram
      ' I need an extra "0" for this line item to keep everything in line
      If histogramIndex < 10 Then
         inOutFile.Write "0"
      End If

      ' I need an extra "0" for this line item to keep everything in line
      If histogramIndex < 100 Then
         inOutFile.Write "0"
      End If

      ' output the index to file
      inOutFile.Write histogramIndex

      ' some spacing to make it look nice
      inOutFile.Write " "

      'figure out how many X's I need
      outputX = CDbl(hist) / largestCount * 100

      'output the X's
      For a = 0 to outputX ' (outputX - 1)
         inOutFile.Write "X"
      Next

      inOutFile.WriteLine

      histogramIndex = histogramIndex + 1

   Next

   inOutFile.WriteLine

End Function

' Function to active all the channels according to the document's mode
' Takes a document reference for input
Private Function TurnOnDocumentHistogramChannels (inDocument)
   ' see how many channels we need to activate
   visibleChannelCount = 0

   'based on the mode of the document
   Select Case inDocument.Mode
      Case 1
         visibleChannelCount = 1
      Case 5
         visibleChannelCount = 1
      Case 6
         visibleChannelCount = 1
      Case 8
         visibleChannelCount = 2
      Case 2
```

```
            visibleChannelCount = 3
        Case 4
            visibleChannelCount = 3
        Case 3
            visibleChannelCount = 4
        Case 8
            visibleChannelCount = 4
        Case 7
            visibleChannelCount = (inDocument.Channels.Count + 1)
        Case Else
            visibleChannelCount = (inDocument.Channels.Count + 1)
    End Select

    ' now get the channels to activate into a local array
    ReDim aChannelArray(visibleChannelCount)

    ' index for the active channels array
    aChannelIndex = 1
    For channelIndex = 1 to inDocument.channels.Count
        If channelIndex <= visibleChannelCount Then
            Set aChannelArray(aChannelIndex) = inDocument.Channels(channelIndex)
            aChannelIndex = aChannelIndex + 1
        End If
    Next

End Function


Private Function StandardDeviationHistogram(inputArray)
    Dim numPixels, sum1, sum2, x, gray

    numPixels = 0
    sum1 = 0.0
    sum2 = 0.0

    ' Compute totals for the various statistics
    For gray = 0 To 255
        x = inputArray(gray)
        numPixels = numPixels + x
        sum1 = sum1 + x * gray
        sum2 = sum2 + x * (gray * gray)
    Next

    StandardDeviationHistogram = Sqr((sum2 - (sum1 * sum1) / numPixels) / (numPixels -
1))

End Function

Private Function AverageHistogram(inputArray)
    Dim numPixels, sum1, sum2, x, gray

    numPixels = 0
    sum1 = 0.0
    sum2 = 0.0

    ' Compute totals for the various statistics
    For gray = 0 To 255
        x = inputArray(gray)
        numPixels = numPixels + x
        sum1 = sum1 + x * gray
```

```
        sum2 = sum2 + x * (gray * gray)
    Next

    AverageHistogram =  sum1 / numPixels

End Function

Private Function MedianHistogram(inputArray, numPixels)
    Dim gray, total, mid

    gray = 0
    total = inputArray(0)
    mid = (numPixels + 1) / 2

    Do While (total < mid)
        gray  = gray + 1
        total = total + inputArray(gray)
    Loop

    MedianHistogram = gray

End Function
```

CMYKColor

The definition of a CMYK color.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Black	Number (Double)	Read-write. The black color value (as percent) (0.0 - 100.0).
Cyan	Number (Double)	Read-write. The cyan color value (as percent) (0.0 - 100.0).
Magenta	Number (Double)	Read-write. The magenta color value (as percent) (0.0 - 100.0).
typename	String	Read-only. The class name of the referenced CMYKColor object.
Yellow	Number (Double)	Read-write. The yellow color value (as percent) (0.0 - 100.0).

ContactSheetOptions

Options that can be specified for a contact sheet.

Properties

Property	Value Type	What it is
AcrossFirst	Boolean	Read-write. Indication of whether to place the images horizontally (left to right, then top to bottom) first. Default: `true`.
BestFit	Boolean	Read-write. Indication of whether to rotate images for the best fit. Default: `false`.
Caption	Boolean	Read-write. Indication of whether to use the filename as a caption for the image. Default: `true`.
ColumnCount	Number (Long)	Read-write. The number of columns to include (1 - 100; default: `5`).
Flatten	Boolean	Read-write. Indication of whether to flatten all layers in the final document. Default: `true`.
Font	PsGalleryFontType	Read-write. The font used for the caption. Default: `1`.
FontSize	Number (Long)	Read-write. The font size to use for the caption. Default: `12`.
Height	Number (Long)	Read-write. The height (in pixels) of the resulting document (100 - 2900; default: `720`).
Horizontal	Number (Long)	Read-write. The horizontal spacing (in pixels) between images (0 - 29000; default: `1`).
Mode	PsNewDocumentMode	Read-write. The document color mode. Default: `2 (psNewRGB)`.
Resolution	Number (Double)	Read-write. The resolution of the document in pixels per inch (35 - 1200; default: `72.0`).
RowCount	Number (Long)	Read-write. The number of rows to use (1 - 100; default: `6`).
typename	String	Read-only. The class name of the referenced `contactSheetOptions` object.
UseAutoSpacing	Boolean	Read-write. Indication of whether to auto space the images. Default: `true`.
Vertical	Number (Long)	Read-write. The vertical spacing (in pixels) between images (0 - 29000; default: `1`). **Note:** Valid only when `UseAutoSpacing = false`.
Width	Number (Long)	Read-write. The width (in pixels) of the resulting document (100 - 2900; default: `576`).

DCS1_SaveOptions

Options that can be specified when saving a CMYK document in DCS1 format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
DCS	PsDCSType	Read-write. Default: 3.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document
Encoding	PsSaveEncoding	Read-write. The type of encoding to use for document. Default: 1.
HalftoneScreen	Boolean	Read-write. Indication of whether to include halftone screen. Default: false.
Interpolation	Boolean	Read-write. Indication of use image interpolation. Default: false)
Preview	PsPreviewType	Read-write. The type of preview. Default: 3.
TransferFunction	Boolean	Read-write. Indication of whether to include the Transfer functions to compensate for dot gain between the image and film. Default: false.
typename	String	Read-only. The class name of the referenced DCS1_SaveOptions object.
VectorData	Boolean	Read-write. Indication of whether to include vector data. **Note:** Valid only if the document includes vector data (un-rasterized text).

DCS2_SaveOptions

Options that can be specified when saving a CMYK document in DCS2 format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
DCS	PsDCSType	Read-write. The type of composite file to create. Default: 1.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
Encoding	PsSaveEncoding	Read-write. The type of encoding to use. Default: 1.
HalftoneScreen	Boolean	Read-write. Indication of whether to include the halftone screen. Default: false.
Interpolation	Boolean	Read-write. Indication of whether to use image interpolation. Default: false.
MultiFileDCS	Boolean	Read-write. Indication of whether to save color channels as multiple files or a single file. Default: false.
Preview	PsPreviewType	Read-write. The preview type. Default: 3.
SpotColors	Boolean	Read-write. Indication of whether to save spot colors.
TransferFunction	Boolean	Read-write. Indication of whether to include the Transfer functions to compensate for dot gain between the image and film. Default: false.
typename	String	Read-only. The class name of the referenced DCS2_SaveOptions object.
VectorData	Boolean	Read-write. Indication of whether to include vector data. **Note:** Valid only if the document includes vector data (un-rasterized text).

Document

The active containment object for layers and all other objects in the script; the basic canvas for the file.

Note: In Adobe Photoshop CS2, a document can also be referred to as an image or a canvas.

- The term *image* refers to the entire document and its contents. You can trim or crop an image. You resize an image using the `ResizeImage()` method.
- The term *canvas* refers to the space in which the document sits on the screen. You can rotate or flip the canvas. You resize the canvas using the `ResizeCanvas()` method.

Properties

Property	Value Type	What it is
ActiveChannels	Array (Channel objects)	Read-write. The selected channels.
ActiveHistoryBrushSource	Object (HistoryState)	Read-write. The history state to use with the history brush.
ActiveHistoryState	Object (HistoryState)	Read-write. The selected `HistoryState` object.
ActiveLayer	Object (Layer)	Read-write. The selected layer.
Application	Object (Application)	Read-only. The application that the collection belongs to.
ArtLayers	Object (ArtLayers)	Read-only. The `ArtLayers` collection.
BackgroundLayer	Object (ArtLayer)	Read-only. Indicates whether the layer is a background layer.
BitsPerChannel	PsBitsPerChannelType	Read-write. The number of bits per channel.
Channels	Object (Channels)	Read-write. The `Channels` collection.
ColorProfileName	String	Read-write. The name of the color profile. **Note:** Valid only when `ColorProfileType = 3` or `ColorProfileType = 2`. See ColorProfileType.
ColorProfileType	PsColorProfileType	Read-write. The type of color model that defines the document's working space.
ComponentChannels	Array (Channel objects)	Read-only. A list of the component color channels.
FullName	Strong	Read-only. The full path name of the document.
Height	Number (Long)	Read-only. The height of the document (unit value).

Property	Value Type	What it is (Continued)
Histogram	Array of 256 Numbers (Long)	Read-only. A histogram showing the number of pixels at each color intensity level for the composite channel. **Note:** Valid only when Mode = 2; Mode = 3; or Mode = 6. See Mode.
HistoryStates	Object (HistoryStates)	Read-only. The HistoryStates collection.
Info	Object (DocumentInfo)	Read-only. Metadata about the document.
LayerComps	Object (LayerComps)	Read-only. The LayerComps collection.
Layers	Object (Layers)	Read-only. The Layers collection.
LayerSets	Object (LayerSets)	Read-only. The LayerSets collection.
Managed	Boolean	Read-only. Indicates whether the document a is workgroup document.
Mode	PsDocumentMode	Read-only. The color profile.
Name	String	Read-only. The document's name.
Parent	Object (Application)	Read-only. The Document object's container.
Path	String	Read-only. The path to the document.
PathItems	Object (PathItems)	Read-only. The PathItems collection.
PixelAspectRatio	Number (Double)	Read-write. The (custom) pixel aspect ratio to use (0.100 - 10.000).
QuickMaskMode	Boolean	Read-write. Indicates whether the document is in Quick Mask mode.
Resolution	Number (Double)	Read-only. The document's resolution (in pixels per inch).
Saved	Boolean	Read-only. Indicates whether the document has been saved since the last change.
Selection	Object (Selection)	Read-only. The selected area of the document.
typename	String	Read-only. The class name of the Document object.
Width	Number (Double)	Read-only. The width of the document (unit value).
XMPMetadata	Object (XMPMetadata)	Read-only. Camera raw settings for the image. **Note:** Valid only for documents opened in Camera Raw format.

Methods

Method	Parameter Type	Returns	What it does
ChangeMode (DestinationMode [, Options])	PsChangeMode object (BitmapConversionOptions or IndexedConversionOptions)		Changes the color profile.
Close ([Saving])	PsSaveOptions		Closes the document. When the optional parameter is used, the document. Default: 3 (psPromptToSaveChanges).
ConvertProfile (DestinationProfile, Intent [, BlackPointCompensation] [, Dither])	String PsIntent Boolean Boolean		Changes the color profile. **Note:** The DestinationProfile parameter must be either a string that names the color mode or Working RGB, Working CMYK, Working Gray, Lab Color (meaning one of the working color spaces or Lab color)
Crop (Bounds [, Angle] [, Width] [, Height])	Array(UnitValue) Number (Double) UnitValue UnitValue		Crops the document. The first parameter is an array of four coordinates that mark the portion remaining after cropping, in the following order: left, top, right, bottom.
Duplicate ()		Document	Creates a duplicate of the Document object.
ExportDocument (ExportIn [, ExportAs] [, Options])	File PsExportType ExportOptionsIllustrator		Exports the document.
Flatten ()			Flattens all layers.
FlipCanvas (Direction)	PsDirection		Flips the image within the canvas in the specified direction.
ImportAnnotations (File)	String		Imports annotations into the document.
MergeVisibleLayers ()			Flattens all visible layers in the document.

Method	Parameter Type	Returns	What it does (Continued)
Paste ([IntoSelection])	Boolean	ArtLayer	Pastes the contents of the clipboard into the document. If the optional argument is set to true and a selection is active, the contents are pasted into the selection.
PrintOut ([PostScriptEncoding] [, SourceSpace] [, PrintSpace] [, Intent] [BlackPointCompensation])	PsPrintEncoding PsSourceSpaceType String PsIntent Boolean		Prints the document. **Note:** PrintSpace specifies the color space for the printer. Valid values are nothing (that is, the same as the source); or Working RGB, Working CMYK, Working Gray, Lab Color (meaning one of the working color spaces or Lab color); or a string specifying a specific colorspace. Default: ***nothing***).
RasterizeAllLayers ()			Rasterizes all layers.
ResizeCanvas ([Width] [, Height] [, Anchor])	UnitValue UnitValue PsAnchorPosition		Changes the size of the canvas to display more or less of the image but does not change the image size. See ResizeImage.
ResizeImage ([Width] [, Height] [, Resolution] [, ResampleMethod])	UnitValue UnitValue Number (Double) PsResampleMethod		Changes the size of the image.
RevealAll ()			Expands the document to show clipped sections.
RotateCanvas (Angle)	Number (Double)		Rotates the canvas (including the image) in clockwise direction.
Save ()			Saves the document.

Method	Parameter Type	Returns	What it does (Continued)
SaveAs (SaveIn [, Options] [, AsCopy] [, ExtensionType])	File object (corresponding SaveOptions object*) Boolean PsExtensionType * Examples: BMPSaveOptions DCS2_SaveOptions JPEGSaveOptions TiffSaveOptions etc.		Saves the document with specified save options. **Note:** The Options parameter's value can be a value from the PsSaveDocumentType constant list, or any of the "***SaveOptions***" objects in the current chapter such as BMPSaveOptions, EPSSaveOptions, JPEGSaveOptions, and so on.
SplitChannels ()		Array (Document objects)	Splits the document channels into separate images.
Trap (Width)	Number (Long)		Applies trapping to a CMYK document. **Note:** Valid only when Mode = 3. See Mode.
Trim ([Type] [, Top] [, Left] [, Bottom] [, Right])	PsTrimType Boolean Boolean Boolean Boolean		Trims the transparent area around the image on the specified sides of the canvas. **Note:** Default is true for all Boolean values.

Sample Script

The following script creates a document that contains two images (an eagle and a duck) obtained from the Adobe Photoshop CS2 Samples folder and employs the following steps:

1. Determine which image is larger.
2. Resize the smaller image to match the larger image.
3. Create a merged document twice as high as either image in order to hold both images.
4. Select part of the document and paste the eagle into the selection. T
5. Invert the selection and paste the duck into the lower part of the document.
6. Position the eagle over the duck.

Document.vbs

```
Dim appRef, startRulerUnits, startTypeUnits, startDisplayDialogs, eagleDoc
Dim duckDoc, mergedDoc, selRegion
```

```
Set appRef = CreateObject("Photoshop.Application")

' Save the current preferences
startRulerUnits = appRef.Preferences.RulerUnits
startTypeUnits = appRef.Preferences.TypeUnits
startDisplayDialogs = appRef.DisplayDialogs

' Set Photoshop CS2 to use pixels and display no dialogs
appRef.Preferences.RulerUnits = 1 'for PsUnits --> 1 (psPixels)
appRef.Preferences.TypeUnits = 1 'for PsTypeUnits --> 1 (psPixels)
appRef.DisplayDialogs = 3 'for PsDialogModes --> 3 (psDisplayNoDialogs)

' first close all the open documents
Do While appRef.Documents.Count
   appRef.ActiveDocument.Close
Loop

' Open the eagle and duck files from the samples folder
Set eagleDoc = appRef.Open(appRef.Path + "/Samples/Eagle.psd")
Set duckDoc = appRef.Open(appRef.Path + "/Samples/Ducky.tif")

' Find out which document is larger
' Resize the smaller document the to the larger document's size
' The resize requires the document be the active/front document
   If (eagleDoc.Width * eagleDoc.Height) > (duckDoc.Width * duckDoc.Height) Then
      appRef.ActiveDocument = duckDoc
      duckDoc.Resize eagleDoc.Width, eagleDoc.Height
   Else
      appRef.ActiveDocument = eagleDoc
      eagleDoc.ResizeImage duckDoc.Width, duckDoc.Height
End If

' Create a new document twice as high as two files
Set mergedDoc = appRef.Documents.Add(duckDoc.Width, duckDoc.Height * 2,
duckDoc.Resolution, "EagleOverDuck")

' Copy the eagle to the top; make it the active document so we can manipulate it
appRef.ActiveDocument = eagleDoc
eagleDoc.ActiveLayer.Copy

'Paste the eagle to the merged document, making the merged document active
appRef.ActiveDocument = mergedDoc

' Select a square area at the top of the new document
selRegion = Array(Array(0, 0), _
              Array(mergedDoc.Width, 0), _
              Array(mergedDoc.Width, mergedDoc.Height / 2), _
              Array(0, mergedDoc.Height / 2), _
              Array(0, 0))

' Create the selection
mergedDoc.Selection.Select(selRegion)

'Paste in the eagle
mergedDoc.Paste

' do the same thing for the duck
appRef.ActiveDocument = duckDoc
duckDoc.ActiveLayer.Copy
```

```
appRef.ActiveDocument = mergedDoc
mergedDoc.Selection.Select(selRegion)

' Inverting the selection so the bottom of the document is now selected
mergedDoc.Selection.Invert

' Paste the duck
mergedDoc.Paste

' get rid of our originals without modifying them
duckDoc.Close( 2) 'for PsSaveOptions --> 2 (psDoNotSaveChanges)
eagleDoc.Close( 2)

' Reset the application preferences
appRef.Preferences.RulerUnits = startRulerUnits
appRef.Preferences.TypeUnits = startTypeUnits
appRef.DisplayDialogs = startDisplayDialogs
```

DocumentInfo

Metadata about a `Document` object. These values can be set by choosing File > File Info in the Adobe Photoshop CS2 application.

Note: You use the object name `Info`, rather than `DocumentInfo`, in a script, as in the following sample, which sets the `Author`, `Caption`, and `Copyrighted` properties:

```
Dim docRef
docRef = Open(fileList[i])
' set the file info
docRef.Info.Author = "Mr. Adobe Programmer"
docRef.Info.Caption = "Adobe photo shoot"
docRef.Info.Copyrighted = 1
```

The following sample uses the DocumentInfo object incorrectly:

```
docRef.DocumentInfo.Author = "Mr. Adobe Programmer"
docRef.DocumentInfo.Caption = "Adobe photo shoot"
docRef.DocumentInfo.Copyrighted = 1
```

Properties

Property	Value Type	What it is
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Author	`String`	Read-write.
authorPosition	`String`	Read-write.
Caption	`String`	Read-write.
CaptionWriter	`String`	Read-write.
Category	`String`	Read-write.
City	`String`	Read-write.
Copyrighted	PsCopyrightedType	Read-write. The copyrighted status.
CopyrightNotice	`String`	Read-write.
Country	`String`	Read-write.
CreationDate	`String`	Read-write.
Credit	`String`	Read-write.
EXIF	`Array of arrays: Array(Array (tag, tag data)), ...)`	Read-only. Camera data that includes camera settings used when the image was taken. Sample array values are: tag = “camera”; tag value = “Cannon”.
Headline	`String`	Read-write.
Instructions	`String`	Read-write.
JobName	`String`	Read-write.

Property	Value Type	What it is (Continued)
Keywords	Array (Strings)	Read-write. A list of keywords that can identify the document or its contents.
OwnerUrl	String	Read-write.
Parent	Object (Document)	Read-only. The Info object's container.
ProvinceState	String	Read-write.
Source	String	Read-write.
SupplementalCategories	Array (Strings)	Read-write.
Title	String	Read-write.
TransmissionReference	String	Read-write.
typename	String	Read-only. The class name of the referenced Info object.
Urgency	PsUrgency	Read-write.

Sample Script

The following script checks to see if any documents are open. If none are open, it opens the sample file eagle.psd.

The script then sets the following document info (metadata):

- Author: Mr. Adobe programmer
- Caption: Adobe Photo shoot
- CaptionWriter: Mr. Adobe programmer
- City: San Jose
- CopyrightNotice: Copyright (c) Adobe Programmer Photography
- Copyrighted status: Copyrighted Work
- Country: USA
- State: CA

Note: After the script finishes running, choose File > File Info to display the metadata set by the script.

DocumentInfo.vbs

```
Dim appRef, startRulerUnits, startTypeUnits, startDisplayDialogs, docRef

Set appRef = CreateObject("Photoshop.Application")

' Save the current preferences
startRulerUnits = appRef.Preferences.RulerUnits
startTypeUnits = appRef.Preferences.TypeUnits
startDisplayDialogs = appRef.DisplayDialogs

' Set Photoshop CS2 to use pixels and display no dialogs
appRef.Preferences.RulerUnits = 1 'for PsUnits --> 1 (psPixels)
appRef.Preferences.TypeUnits = 1 'for PsTypeUnits --> 1 (psPixels)
```

```
appRef.DisplayDialogs = 3 'for PsDialogModes --> 3 (psDisplayNoDialogs)

' if there are no documents open then try to open a sample file
If appRef.Documents.Count = 0 Then
   appRef.Open(appRef.Path + "/Samples/Eagle.psd")
End If

Set docRef = appRef.ActiveDocument

' tag all of the documents with photo shoot information
docRef.Info.Author = "Mr. Adobe Programmer"
docRef.Info.Caption = "Adobe Photo shoot"
docRef.Info.captionWriter = "Mr. Adobe Programmer"
docRef.Info.city = "San Jose"
docRef.Info.CopyrightNotice = "Copyright (c) Adobe Programmer Photography"
docRef.Info.Copyrighted = 1 'for PsCopyrightedType --> 1 (psCopyrightedWork)
docRef.Info.country = "USA"
docRef.Info.provinceState = "CA"

' Reset the application preferences
appRef.DisplayDialogs = startDisplayDialogs
appRef.Preferences.RulerUnits = startRulerUnits
appRef.Preferences.TypeUnits = startTypeUnits
```

Documents

The collection of open `Document` objects.

Note: See 'Document' on page 51 for information on the `Document` object.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `Documents` collection.
Item	Object (Document)	Read-only. Gets an element from the collection.
Parent	Object (Application)	Read-only. The `Documents` objects' container.
typename	String	Read-only. The class name of the referenced `Documents` object.

Methods

Method	Parameter Type	Returns	What it does
Add ([Width] [, Height] [, Resolution] [, Name] [, Mode] [, InitialFill] [PixelAspectRatio])	UnitValue UnitValue Number (Double) String PsNewDocumentMode PsDocumentFill Number (Double)	Document	Adds a `Document` Object (`PixelAspectRatio`: 0.100 0 10.00).
Index (ItemPtr)	object (Document)	Number (Long)	Gets an element from the `Documents` collection.

EPSOpenOptions

Options that can be specified when opening an EPS format document.

Properties

Property	Value Type	What it is
AntiAlias	Boolean	Read-write. Indication of whether to use antialias.
Application	Object (Application)	Read-only. The application that the collection belongs to.
ConstrainProportions	Boolean	Read-write. Indication of whether to constrain the proportions of the image.
Height	Number (Double)	Read-write. The height of the image (unit value).
Mode	PsOpenDocumentMode	Read-write. The color profile to use as the document mode.
Resolution	Number (Double)	Read-write. The resolution of the document in pixels per inch.
typename	String	Read-only. The class name of the referenced EPSOpenOptions object.
Width	Number (Double)	Read-write. The width of the image (unit value).

EPSSaveOptions

Options that can be specified when saving a document in EPS format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in this document.
Encoding	PsSaveEncoding	Read-write. The type of encoding to use. Default: 1.
HalftoneScreen	Boolean	Read-write. Indication of whether to include the halftone screen. Default: false.
Interpolation	Boolean	Read-write. Indication of whether to use image interpolation. Default: false.
Preview	PsPreviewType	Read-write. The preview type.
PsColorManagement	Boolean	Read-write. Indication of whether to use Postscript color management. Default: false.
TransferFunction	Boolean	Read-write. Indication of whether to include the Transfer functions to compensate for dot gain between the image and film. Default: false.
TransParentWhites	Boolean	Read-write. Indication of whether to display white areas as transParent. **Note:** Valid only when *Document*.Mode = 5. See 'Mode' on page 52 (in the Properties table of the Document object) or 'ChangeMode' on page 53 (in the Methods table of the Document object).
typename	String	Read-only. The class name of the referenced EPSSaveOptions object.
VectorData	Boolean	Read-write. Indication of whether to include vector data. **Note:** Valid only if the document includes vector data (text).

ExportOptionsIllustrator

Options that can be specified when exporting a PathItem object to an Adobe Illustrator® file.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Path	PsIllustratorPathType	Read-write. The type of path to export. Default: 1.
PathName	String	Read-write. The name of the path to export. **Note:** Valid only when Path = 3. See Path.
typename	String	Read-only. The class name of the referenced ExportOptionsIllustrator object.

ExportOptionsSaveForWeb

Options that can be specified when optimizing a document for the Web.

Properties

Property	Value type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Blur	Number (Double)	Read-write. Applies blur to the image to reduce artifacts. Default: 0.0.
ColorReduction	PsColorReductionType	Read-write. The color reduction algorithm. Default: 1 (psSelective).
Colors	Number (Long)	Read-write. The number of colors in the palette. Default: 256.
Dither	PsDitherType	Read-write. The type of dither. Default: 2 (psDiffusion).
DitherAmount	Number (Long)	Read-write. The amount of dither. Default: 100. **Note:** Valid only when Dither = 2. See Dither.
Format	PsSaveDocumentType	Read-write. The file format to use. Default: 3 (psCompuServeGIFSave).
IncludeProfile	Boolean	Read-write. Indication of whether to include the document's embedded color profile. Default: false.
Interlaced	Boolean	Read-write. Indication of whether to download in multiple passes; progressive. Default: false.
Lossy	Number (Long)	Read-write. The amount of lossiness allowed. Default: 0.
MatteColor	RGBColor	Read-write. The colors to blend transParent pixels against.
Optimized	Boolean	Read-write. Indication of whether to create smaller but less compatible files. Default: true. **Note:** Valid only when format = 6 (psJPEGSave). See Format.
PNG8	Boolean	Read-write. Indicates the number of bits; true = 8, false = 24. Default: true. **Note:** Valid only when format = 13 (psPNGSave). See Format.
Quality	Number (Long)	Read-write. The quality of the produced image (0 - 100 as percentage; default: 60).

Property	Value type	What it is (Continued)
Transparency	Boolean	Read-write. Indication of transParent areas of the image should be included in the saved image. Default: `true`.
TransparencyAmount	Number (Long)	Read-write. The amount of transparency dither. Default: 100. **Note:** Valid only if `Transparency = true`. See Transparency.
TransparencyDither	PsDitherType	Read-write. The transparency dither algorithm. Default: `1`.
typename	String	Read-only. The class name of the referenced `ExportOptionsSaveForWeb` object.
WebSnap	Number (Long)	Read-write. The tolerance amount within which to snap close colors to Web palette colors. Default: `0`.

GalleryBannerOptions

Options that define the `BannerOptions` property of the `GalleryOptions` object. See 'GalleryOptions' on page 71.

Tip: You can preserve default values for many `GalleryBannerOptions` properties by setting the `GalleryOptions` property `PreserveAllMetadata` to `true` or by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** on the Options area of the Web Photo Gallery dialog.

Properties

Property	Value Type	What it is
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
ContactInfo	`String`	Read-write. The Web photo gallery contact info.
Date	`String`	Read-write. The Web photo gallery date. Default: current date.
Font	`PsGalleryFontType`	Read-write. The font setting for the banner text. Default: `1`.
FontSize	`Number (Long)`	Read-write. The font size for the banner text (1 - 7; default: `3`).
Photographer	`String`	Read-write. The Web photo gallery photographer.
SiteName	`String`	Read-write. The Web photo gallery site name. Default: `Adobe Web Photo Gallery`.
typename	`String`	Read-only. The class name of the referenced `GalleryBannerOptions` object.

GalleryCustomColorOptions

Options that define the `customColorOptions` property of the `GalleryOptions` object. See 'GalleryOptions' on page 71.

Tip: You can preserve default values for many `GalleryCustomColorOptions` properties by setting the `GalleryOptions` property `PreserveAllMetadata` to `true` or by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** on the Options area of the Web Photo Gallery dialog.

Properties

Property	Value Type	What it is
ActiveLinkColor	Object (RGBColor)	Read-write. The color to use to indicate an active link.
Application	Object (Application)	Read-only. The application that the collection belongs to.
BackgroundColor	Object (RGBColor)	Read-write. The background color.
BannerColor	Object (RGBColor)	Read-write. The banner color.
LinkColor	Object (RGBColor)	Read-write. The color to use to indicate a link.
TextColor	Object (RGBColor)	Read-write. The text color.
typename	String	Read-only. The class name of the referenced `GalleryCustomColorOptions` object.
VisitedLinkColor	Object (RGBColor)	Read-write. The color to use to indicate a visited link.

GalleryImagesOptions

Options that define the `ImagesOptions` property of the `GalleryOptions` object. See 'GalleryOptions' on page 71.

Tip: You can preserve default values for many `GalleryImagesOptions` properties by setting the `GalleryOptions` property `PreserveAllMetadata` to `true` or by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** on the Options area of the Web Photo Gallery dialog.

Properties

Property	Value Type	What it is
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Border	`Number (Long)`	Read-write. The size (in pixels) of the border that separates images (0 - 99; default: `0`).
Caption	`Boolean`	Read-write. Indication of whether to generate image captions. Default: `false`.
Dimension	`Number (Long)`	Read-write. The resized image dimensions in pixels. Default: `350`. **Note:** Valid only when `ResizeImages = true`. See ResizeImages.
Font	`PsGalleryFontType`	Read-write. The font to use for image captions. Default: `1`.
FontSize	`Number (Long)`	Read-write. The font size for image captions (1 - 7; default: `3`). **Note:** Valid only when `Caption = true`. See Caption.
ImageQuality	`Number (Long)`	Read-write. The quality setting for a JPEG image (0 - 12; default: `5`).
IncludeCopyright	`Boolean`	Read-write. Indication of whether to include copyright information in captions. Default: `false`. **Note:** Valid only when `Caption = true`. See Caption.
IncludeCredits	`Boolean`	Read-write. Indication of whether to include the credits in image captions. Default: `false`. **Note:** Valid only when `Caption = true`. See Caption.

Property	Value Type	What it is (Continued)
IncludeFilename	Boolean	Read-write. Indication of whether to include the file name in image captions. Default: true. **Note:** Valid only when Caption = true. See Caption.
IncludeTitle	Boolean	Read-write. Indication of whether to include the title in image captions. Default: false. **Note:** Valid only when Caption = true. See Caption.
NumericLinks	Boolean	Read-write. Indication of whether to add numeric links. Default: true.
ResizeConstraint	PsGalleryConstrainType	Read-write. The image dimensions to constrain in the gallery image. Default: 3. **Note:** Valid only when ResizeImages = true. See ResizeImages.
ResizeImages	Boolean	Read-write. Indication of whether to automatically resize images for placement on the gallery pages. Default: true.
typename	String	Read-only. The class name of the referenced GalleryImagesOptions object.

GalleryOptions

Options that can be specified for a Web photo gallery.

Tip: You can preserve default values for many `GalleryOptions` properties by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** on the Options area of the Web Photo Gallery dialog**.**

Properties

Property	Value Type	What it is
AddSizeAttributes	`Boolean`	Read-write. Indicates whether width and height attributes for images will be added. Default: `true`.
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
BannerOptions	`Object (GalleryBannerOptions)`	Read-write. The options related to banner settings.
CustomColorOptions	`Object (GalleryCustomColorOptions)`	Read-write. The options related to custom color settings.
EmailAddress	`String`	Read-write. The email address to show on the Web page.
ImagesOptions	`Object (GalleryImagesOptions)`	Read-write. The options related to images settings.
IncludeSubFolders	`Boolean`	Read-write. Indication of whether to include all files found in sub folders of the input folder. Default: `true`.
LayoutStyle	`String`	Read-write. The style to use for laying out the Web page. Default: `Centered Frame 1 - Basic`.
PreserveAllMetadata	`Boolean`	Read-write. Indicates whether to save metadata. Default: `false`.
SecurityOptions	`Object (GallerySecurityOptions)`	Read-write. The options related to security settings.
ThumbnailOptions	`Object (GalleryThumbnailOptions)`	Read-write. The options related to thumbnail image settings.
typename	`String`	Read-only. The class name of the referenced `GalleryOptions` object.

Property	Value Type	What it is (Continued)
UseShortExtension	Boolean	Read-write. Indicates whether the short Web page extension .htm or Number (Long) Web page extension .html will be used. Default: true.
UseUTF8Encoding	Boolean	Read-write. Indicates whether the Web page should use UTF-8 encoding. Default: false.

GallerySecurityOptions

Options that define the `SecurityOptions` property of the `GalleryOptions` object. See 'GalleryOptions' on page 71.

Tip: You can preserve default values for many `GallerySecurityOptions` properties by setting the `GalleryOptions` property `PreserveAllMetadata` to `true` or by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** on the Options area of the Web Photo Gallery dialog.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Content	PsGallerySecurityType	Read-write. The Web photo gallery security content. Default: 1.
Font	PsGalleryFontType	Read-write. The Web photo gallery security font. Default: 1.
FontSize	Number (Long)	Read-write. The Web photo gallery security font size (1 - 72; default: 3).
Opacity	Number (Long)	Read-write. The Web page security opacity as a percent. Default: 100.
Text	String	Read-write. The Web photo gallery security custom text.
TextColor	Object (RGBColor)	Read-write. The Web page security text color.
TextPosition	PsGallerySecurityTextPositionType	Read-write. The Web photo gallery security text position. Default: 1.
TextRotate	PsGallerySecurityTextRotateType	Read-write. The Web photo gallery security text orientation to use. Default: 1.
typename	String	Read-only. The class name of the referenced `GallerySecurityOptions` object.

GalleryThumbnailOptions

Options that define the `thumbnailOptions` property of the `GalleryOptions` object. See 'GalleryOptions' on page 71.

Tip: You can preserve default values for many `GalleryThumbnailOptions` properties by setting the `GalleryOptions` property `PreserveAllMetadata` to `true` or by choosing **File > Automate > Web Photo Gallery**, and then choosing **Preserve all metadata** on the Options area of the Web Photo Gallery dialog.

Properties

Property	Value Type	What it is
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Border	`Number (Long)`	Read-write. The amount of border pixels you want around your thumbnail images (0 - 99; default: `0`).
Caption	`Boolean`	Read-write. Indicates whether there is a caption. Default: `false`.
ColumnCount	`Number (Long)`	Read-write. The number of columns on the page. Default: `5`.
Dimension	`Number (Long)`	Read-write. The Web photo gallery thumbnail dimension in pixels. Default: `75` .
Font	`PsGalleryFontType`	Read-write. The Web photo gallery font. Default: `1`.
FontSize	`Number (Long)`	Read-write. The font size for thumbnail images text (1 - 7; default: `3`).
IncludeCopyright	`Boolean`	Read-write. Indication of whether to include copyright information for thumbnails. Default: `false`.
IncludeCredits	`Boolean`	Read-write. Indication of whether to include credits for thumbnails. Default: `false`.
IncludeFilename	`Boolean`	Read-write. Indication of whether to include file names for thumbnails. Default: `false`.
IncludeTitle	`Boolean`	Read-write. Indication of whether to include titles for thumbnails. Default: `false`.
RowCount	`Number (Long)`	Read-write. The number of rows on the page. Default: `3`.
Size	`PsGalleryThumbSizeType`	Read-write. The thumbnail image size. Default: `2`.
typename	`String`	Read-only. The class name of the referenced `GalleryThumbnailOptions` object.

GIFSaveOptions

Options that can be specified when saving a document in GIF format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Colors	Number (Long)	Read-write. The number of palette colors. **Note:** Valid only when Palette = 2; Palette = 3; Palette = 4; Palette = 5; Palette = 6; Palette = 7; or Palette = 8. See Palette.
Dither	PsDitherType	Read-write. The dither type.
DitherAmount	Number (Long)	Read-write. The amount of dither. (1 - 100; default: 75). **Note:** Valid only for when Dither = 2. See Dither.
Forced	PsForcedColors	Read-write. The type of colors to force into the color Palette.
Interlaced	Boolean	Read-write. Indicates whether rows should be interlaced. Default: false.
Matte	PsMatteType	Read-write. The color to use to fill antialiased edges adjacent to transParent areas of the image. Default: 4. **Note:** When Transparency = false, the matte color is applied to transParent areas. See Transparency.
Palette	PsPaletteType	Read-write. The type of palette to use. Default: Palette.7.
PreserveExactColors	Boolean	Read-write. Indication of whether to protect colors in the image that contain entries in the color table from being dithered. **Note:** Valid only when Dither = 2. See Dither.
Transparency	Boolean	Read-write. Indication of whether to preserve transParent areas of the image during conversion to GIF format.
typename	String	Read-only. The class name of the referenced GIFSaveOptions object.

GrayColor

Options for defining a gray color.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Gray	Number (Double)	Read-write. The gray value (0.0 - 100.0; default: 0.0).
typename	String	Read-only. The class name of the referenced grayColor object.

HistoryState

A version of the document stored automatically (and added to the `HistoryStates` collection), which preserves the document's state, each time the document is saved.

Note: See 'HistoryStates' on page 116 for information about the `HistoryStates` collection.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Name	String	Read-only. The `HistoryState` object's name.
Parent	Object (Document)	Read-only. The `HistoryState` object's container.
Snapshot	Boolean	Read-only. Indicates whether the history state is a snapshot.
typename	String	Read-only. The class name of the referenced `HistoryState` object.

HistoryStates

The collection of `HistoryState` objects in the document.

Note: See 'HistoryState' on page 77 for more information on `HistoryState` objects.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `HistoryStates` collection.
Item	Object (HistoryState)	Read-only. Gets an element from the collection.
Parent	Object (Document)	Read-only. The `HistoryStates` object's container.
typename	String	Read-only. The class name of the referenced `HistoryStates` object.

Methods

Method	Parameter Type	Returns	What it does
Index (ItemPtr)	Object (HistoryState)	Number (Long)	Gets an element from the `HistoryStates` collection.

HSBColor

Options that can be specified for a color object using the HSB color model.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Brightness	Number (Double)	Read-write. The brightness value (0.0 - 100.0).
Hue	Number (Double)	Read-write. The hue value (0.0 - 100.0).
Saturation	Number (Double)	Read-write. The saturation value (0.0 - 100.0).
typename	String	Read-only. The class name of the referenced HSBColor object.

IndexedConversionOptions

Options that can be specified when converting an RGB image to an indexed color model.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Colors	Number (Long)	Read-write. The number of palette colors. **Note:** Valid only when `Palette = 2`; `Palette = 3`; `Palette = 4`; `Palette = 5`; `Palette = 6`; `Palette = 7`; or `Palette = 8`. See Palette.
Dither	PsDitherType	Read-write. The dither type.
DitherAmount	Number (Long)	Read-write. The amount of dither. (1 - 100). **Note:** Valid only when `Dither = 2`.
Forced	PsForcedColors	Read-write. The type of colors to force into the color palette.
Matte	PsMatteType	Read-write. Read-write. The color to use to fill antialiased edges adjacent to transParent areas of the image. Default: `4`. **Note:** When `Transparency = false`, the matte color is applied to transParent areas. See Transparency.
Palette	PsPaletteType	Read-write. The palette type. Default: `1`.
PreserveExactColors	Boolean	Read-write.Indication of whether to protect colors in the image that contain entries in the color table from being dithered. **Note:** Valid only when `Dither = 2`. See Dither.
Transparency	Boolean	Read-write. Indication of whether to preserve transParent areas of the image during conversion to GIF format.
typename	String	Read-only. The class name of the referenced `IndexedConversionOptions` object.

JPEGSaveOptions

Options that can be specified when saving a document in JPEG format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
FormatOptions	PsFormatOptionsType	Read-write.The download format to use. Default: 1.
Matte	PsMatteType	Read-write. The color to use to fill antialiased edges adjacent to transParent areas of the image. Default: 4. **Note:** When Transparency = false, the matte color is applied to transParent areas. See Transparency.
Quality	Number (Long)	Read-write. The image quality setting to use (affects file size and compression) (0 - 12; default: 3).
Scans	Number (Long)	Read-write. The number of scans to make to incrementally display the image on the page (3 - 5; default: 3). **Note:** Valid only for when FormatOptions = 3.
typename	String	Read-only. The class name of the referenced JPEGSaveOptions object.

LabColor

Options that can be specified when defining a color object using the LAB color model.

Properties

Property	Value Type	What it is
A	Number (Double)	Read-write. The a-value (-128.0 - 127.0).
Application	Object (Application)	Read-only. The application that the collection belongs to.
B	Number (Double)	Read-write. The b-value (-128.0 - 127.0).
L	Number (Double)	Read-write. The L-value (0.0 - 100.0).
typename	String	Read-only. The class name of the referenced LabColor object.

LayerComp

A snapshot of a state of the layers in a document (can be used to view different page layouts or compositions).

Properties

Property	Value Type	What it is
Appearance	Boolean	Read-write. Indication of whether to use layer appearance (layer styles) settings.
Application	Object (Application)	Read-only. The application that the collection belongs to.
Comment	String	Read-write. A description of the layer comp.
Name	String	Read-write. The name of the layer comp.
Parent	Object (Document)	Read-write. The LayerComp object's container.
Position	Boolean	Read-write. Indication of whether to use layer position.
Selected	Boolean	Read-only. Indication of whether the layer comp is currently selected.
typename	String	Read-only. The class name of the referenced LayerComp object.
Visibility	Boolean	Read-write. Indication of whether to use layer visibility settings.

Methods

Method	Parameter Type	Returns	What it does
Apply ()			Applies the layer comp to the document.
Recapture ()			Recaptures the current layer state(s) for this layer comp.
Remove ()			Deletes the LayerComp object.
ResetfromComp ()			Resets the layer comp state to the document state.

LayerComps

The collection of `LayerComp` objects in the document.

Note: See 'LayerComp' on page 121 for information on `LayerComp` objects.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `LayerComps` collection.
Item	Object (LayerComp)	Read-only. Gets an element from the collection.
Parent	Object (Document)	Read-only. The `LayerComps` object's container.
typename	String	Read-only. The class name of the referenced `LayerComps` object.

Methods

Method	Parameter Type	Returns	What it does
Add (Name, Comment, Appearance, Position, Visibility)	String String Boolean Boolean Boolean	LayerComp	Adds a layer comp.
Index (ItemPtr)	Object (LayerComp)	Number (Long)	Gets an element from the `LayerComps` collection.
RemoveAll ()			Removes all `LayerComp` objects from the `LayerComps` collection.

Layers

The collection of layer objects, including `ArtLayer` and `LayerSet` objects, in the document.

Note: See 'ArtLayer' on page 56 for information on `ArtLayer` objects. See 'LayerSet' on page 124 for information on `LayerSet` objects.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `Layers` collection.
Item	Object (Layer)	Read-only. Gets an element from the collection.
Parent	Object (Document or LayerSet)	Read-only. The `Layers` object's container.
typename	String	Read-only. The class name of the referenced `Layers` object.

Methods

Method	Parameter Type	Returns	What it does
Index	Object (Application)	Number (Long)	Gets an element from the collection.
RemoveAll ()			Removes all layers from the collection.

LayerSet

A group of layer objects, which can include `ArtLayer` objects and other (nested) `LayerSet` objects. A single command or set of commands manipulates all layers in a `LayerSet` object.

Properties

Property	Value Type	What it is
AllLocked	Boolean	Read-write. Indicates whether the contents in the layers contained in the `LayerSet` object are editable.
Application	Object (Application)	Read-only. The application that the collection belongs to.
ArtLayer	Object (ArtLayer)	Read-only. When `LayerType = 1 (psArtLayer)`, this property returns a reference to the corresponding `LayerSet` object. See LayerType.
ArtLayers	Object (ArtLayers)	Read-only. The ArtLayers in this LayerSet.
BlendMode	PsBlendMode	Read-write. The blend mode to use for the layer set.
Bounds	Array(UnitValue)	Read-only. The bounding rectangle of the layer set.
EnabledChannels	Array (Channel objects)	Read-write. The channels enabled for the layer set; must be a list of component channels. **Note:** See `Kind` in the Properties table for the Channel Object ('Channel' on page 40).
Layer	Object (layer)	Read-only. The layer corresponding to the Layer Set.
Layers	Object (Layers)	Read-only. The layers in this `LayerSet` object.
LayerSet	Object (LayerSet)	Read-only. When `LayerType = 2 (psLayerSet)`, this property returns a reference to the corresponding `LayerSet` object. See LayerType.
LayerSets	Object (LayerSets)	Read-only. The top level LayerSets in this document.
LayerType	PsLayerType	Read-only. The type of layer.
LinkedLayers	Array (layers)	Read-only. The layers linked to this `LayerSet` object.
Name	String	Read-write. The name of the `LayerSet` object.
Opacity	Number (Double)	Read-write. The master opacity of the `LayerSet` Object (0.0 - 100.0).
Parent	Object (Document or LayerSet)	Read-only. The `LayerSet` object's container.
typename	String	Read-only. The class name of the referenced `LayerSet` object.
Visible	Boolean	Read-write. Indicates whether the `LayerSet` object is visible.

Methods

Method	Parameter Type	Returns	What it does
Duplicate ([RelativeObject] [, InsertionLocation])	object (ArtLayer or LayerSet) PsElementPlacement	Object (Layer)	Creates a duplicate of the LayerSet object.
Link ()	Object (Application)		Links the layer set with another layer.
Merge ()		ArtLayer	Merges the layerset; returns a reference to the art layer created by this method.
Move (RelativeObject, InsertionLocation)	Object (Application) PsElementPlacement		Moves the LayerSet object.
Remove ()			Deletes the LayerSet object.
Resize ([Horizontal] [, Vertical] [, Anchor])	Number (Double) Number (Double) PsAnchorPosition		Resizes all layers in the layer set to the specified dimensions (as a percentage of its current size) and places the layer set in the specified position.
Rotate (Angle [, Anchor])	Number (Double) PsAnchorPosition		Rotates all layers in the layer set around the specified anchor point.
Translate ([DeltaX] [, DeltaY])	UnitValue UnitValue		Moves the position relative to its current position.
Unlink ()			Unlinks the layer set.

LayerSets

The collection of `LayerSet` objects in the document.

Note: See 'LayerSet' on page 124 for information on `LayerSet` objects.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `LayerSets` collection.
Item	Object (LayerSet)	Read-only. Gets an element from the collection.
Parent	Object (Document or LayerSet)	Read-only. The `LayerSets` object's container.
typename	String	Read-only. The class name of the referenced `LayerSets` object.

Methods

Method	Parameter Type	Returns	What it does
Add ()		LayerSet	Creates a new `LayerSet` object.
Index (ItempTR)	Object (LayerSet)	Number (Long)	Gets an element from the `LayerSets` collection.
RemoveAll ()			Removes the layer set, and any layers or layer sets it contains, from the document.

Sample Script

The following script creates three layer sets, then nests a second layer set in each layer set, and then creates a text layer in each nested set that displays the text "Layer in *n* Set Inside *n* Set", where *n* represents the ordinal number of the set (first, second, or third).

LayerSets.vbs

```
Dim appRef, docRef, myLayerSets(3,3),  textArray, i, myLayers(3)

Set appRef = CreateObject("Photoshop.Application")

'close all open documents
Do While appRef.Documents.Count
   appRef.ActiveDocument.Close
Loop

' create a working document
Set docRef = appRef.Documents.Add
```

```
' Create an array to hold the text
textArray = Array("First", "Second", "Third")

'Create an indexer variable
i = 0

' Create three layer sets at the top level
for i = 0 to 2
   Set myLayerSets(i,0) = docRef.LayerSets.Add
Next

' Rearrange the layer sets with the first one on top, second next, etc.
myLayerSets(1,0).moveAfter(myLayerSets(0,0))
myLayerSets(2,0).moveAfter(myLayerSets(1,0))

' Create a layer set inside each layer set
for i = 0 to 2
   myLayerSets(i,0).Name = textArray(i) + " Set"
   Set myLayerSets(i,1) = myLayerSets(i,0).LayerSets.Add
   myLayerSets(i,1).Name = "Inside " + textArray(i) + " Set"
Next

' Create a text layer with a description inside each layer set
for i = 0 to 2
   Set myLayers(i) = myLayerSets(i,1).ArtLayers.Add
   myLayers(i).Kind = 2 ' PsLayerKind.psTextLayer
   myLayers(i).textItem.Contents = "Layer in " & textArray(i) & " Set Inside " _
                                & textArray(i) & " Set"
   myLayers(i).textItem.position = Array(appRef.ActiveDocument.Width * i * 0.33, _
                                   appRef.ActiveDocument.Height * (i + 1) * 0.25)
   myLayers(i).textItem.Size = 12
Next
```

LensBlurOptions

Defines the optional parameter of the `ArtLayer` object's `ApplyLensBlur()` method.

Note: See `ApplyLensBlur` (in the Methods table of the ArtLayer object).

Properties

Property	Value type	What it is
Amount	`Number (Long)`	Read-write. The amount of noise. Default: `0`.
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
BladeCurvature	`Number (Long)`	Read-write. The blade curvature of the iris. Default: `0`.
Brightness	`Number (Long)`	Read-write. The brightness for the specular highlights. Default: `0`.
Distribution	PsNoiseDistribution	Read-write. The distribution value for the noise. Default: `1 (psUniformNoise)`.
FocalDistance	`Number (Long)`	Read-write. The blur focal distance for the depth map. Default: `0`.
InvertDepthMap	`Boolean`	Read-write. Indicates whether the depth map is inverted. Default: `false`.
Monochromatic	`Boolean`	Read-write. Indicates whether the noise is monochromatic. Default: `false`.
Radius	`Number (Long)`	Read-write. The radius of the iris. Default: `15`.
Rotation	`Number (Long)`	Read-write. The rotation of the iris. Default: `0`.
Shape	PsGeometry	The shape of the iris. Default: `2 (psHexagon)`.
Source	PsDepthMapSource	Read-write. The source for the depth map. Default: `1 (psNoSource)`.
Threshold	`Number (Long)`	Read-write. The threshold for the specular highlights. Default: `0`.
typename	`String`	Read-only. The class name of the referenced `lensBlurOptions` object.

NoColor

An object that represents a missing color.

Properties

Property	Value type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
typename	String	Read-only. The class name of the referenced noColor object.

Notifier

An event-handler object that tells the script to execute specified code when a specified event occurs.

Properties

Property	Value type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Event	String	Read-only. The event ID in four characters or a unique String that the notifier is associated with.
EventClass	String	Read-only. The class ID of the event associated with the Notifier object, four characters or a unique string. **Note:** For a list of four-character codes, see Appendix A: Event ID Codes.
EventFile	File	Read-only. The path to the file to execute when the event occurs/activates the notifier.
Parent	Object (Application)	Read-only. The Notifier object's container.
typename	String	Read-only. The class name of the referenced Notifier object.

Methods

Method	Parameter type	Returns	What it does
Remove ()			Deletes the Notifier object. **Note:** You can remove a Notifier object from the Script Events Manager drop-down list by deleting the file named Script Events Manager.xml from in the Photoshop preferences folder. See Adobe Photoshop CS2 help for more information.

Notifiers

The collection of `Notifier` objects in the document; the `Notifiers` property of the Application object.

Note: See 'Notifier' on page 92 for information on `Notifier` objects. See Notifiers (in the Properties table of the **`Application`** object).

Properties

Property	Value type	What it is
`Application`	`Object` (`Application`)	Read-only. The application that the collection belongs to.
`Count`	`Number (Long)`	Read-only. The number of elements in the `Notifiers` collection.
`EventClass`	`String`	Read-only. The class ID of the event.
`Item`	`Object` (`Notifier`)	Read-only. Gets an element from the collection.
`Parent`	`Object` (`Application`)	Read-only. The `Notifiers` object's container
`typename`	`String`	Read-only. Read-only. The class name of the referenced `Notifiers` object.

Methods

Method	Parameter type	Returns	What it does
Add (Event, EventFile [, EventClass])	String file	Notifier	Creates a Notifier object. **Note:** EventClass defines the class ID of the event: four characters or a unique string . For a list of four-character codes, see Appendix A: Event ID Codes. **Tip:** Remember to omit the single quotes when including a four-character ID in your code. **Note:** An EventClass value corresponds to the value you would type in the Descriptive Lable box when adding an event in the Script Events Manager in the Adobe Photoshop CS2 application. For more information on using the Script Events Manager, please refer to Adobe Photoshop CS2 help.
Index (ItemPtr)	Object (Notifier)	Number (Long)	Gets an element from the Notifiers collection.
RemoveAll ()			Removes all Notifier objects from the Notifiers collection. **Note:** You can remove a notifier object from the Script Events Manager drop-down list by deleting the file named Script Events Manager.xml from in the Photoshop preferences folder. See Adobe Photoshop CS2 help for more information.

PathItem

A path or drawing object, such as the outline of a shape or a straight or curved line, which contains sub paths that comprise its geometry.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Kind	PsPathKind	Read-write. The PathItem object's type.
Name	String	Read-write. The PathItem object's name.
Parent	Object (Document)	Read-only. The PathItem object's container.
SubPathItems	Object (SubPathItems)	Read-only. The sub path objects for this PathItem object.
typename	String	Read-only. The class name of the referenced PathItem object.

Methods

Method	Parameter Type	Returns	What it does
Delete ()			Deletes this PathItem object.
Deselect ()			Deselects this PathItem object.
Duplicate (Name)	String		Duplicates this PathItem object with the new name specified in the argument.
FillPath ([FillColor] [, Mode] [, Opacity] [, PreserveTransparency] [, Feather] [, WholePath] [, AntiAlias])	Object (SolidColor, ArtLayer, HistoryState); or String PsColorBlendMode Number (Double) Boolean Number (Double) Boolean Boolean		Fills the area enclosed by the path (Opacity: 0 - 100 as percent; Feather: 0.0 - 250.0 in pixels).
MakeClippingPath ([Flatness])	Number (Double)		Makes this PathItem object the clipping path for this document; the optional parameter tells the PostScript printer how to approximate curves in the path (0.2 - 100).

Method	Parameter Type	Returns	What it does (Continued)
MakeSelection ([Feather] [, AntiAlias] [, Operation])	Number (Double) Boolean PsSelectionType		Makes a Selection object, whose border is the path, from this PathItem Object (Feather: 0.0 - 250.0 in pixels). **Note:** See 'Selection' on page 123.
Select ()			Makes this PathItem object the active or selected PathItem object.
StrokePath ([Tool] [, SimulatePressure])	PsToolType Boolean		Strokes the path with the specified information.

Sample Script

The following creates a path in three segments: two diagonal lines that form a *V*, and a curved line above the *V* that makes it look like a 2D ice cream cone.

Paths.vbs

```
Dim appRef, startRulerUnits, startTypeUnits, startDisplayDialogs, docRef
Dim lineArray(2), lineArray2(2), lineArray3(3), lineSubPathArray(3), myPathItem

Set appRef = CreateObject("Photoshop.Application")

' Save the current preferences
startRulerUnits = appRef.Preferences.RulerUnits
startTypeUnits = appRef.Preferences.TypeUnits
startDisplayDialogs = appRef.DisplayDialogs

' Set Photoshop CS2 to use pixels and display no dialogs
appRef.Preferences.RulerUnits = 1 'for PsUnits --> 1 (psPixels)
appRef.Preferences.TypeUnits = 1 'for PsTypeUnits --> 1 (psPixels)
appRef.DisplayDialogs = 3 'for PsDialogModes --> 3 (psDisplayNoDialogs)

' first close all the open documents
Do While appRef.Documents.Count
   appRef.ActiveDocument.Close
Loop

' create a document to work with
Set docRef = appRef.Documents.Add(5000, 7000, 72, "Simple Line")

'line #1--it's a straight line so the coordinates for anchor, left, and
'right for each point have the same coordinates
Set lineArray(1) = CreateObject("Photoshop.PathPointInfo")
lineArray(1).Kind = 2 ' for PsPointKind --> 2 (psCornerPoint)
lineArray(1).Anchor = Array(100, 100)
lineArray(1).LeftDirection = lineArray(1).Anchor
lineArray(1).RightDirection = lineArray(1).Anchor
```

```
Set lineArray(2) = CreateObject("Photoshop.PathPointInfo")
lineArray(2).Kind = 2
lineArray(2).Anchor = Array(150, 200)
lineArray(2).LeftDirection = lineArray(2).Anchor
lineArray(2).RightDirection = lineArray(2).Anchor

Set lineSubPathArray(1) = CreateObject("Photoshop.SubPathInfo")
lineSubPathArray(1).operation = 2 'for PsShapeOperation --> 2 (psShapeXOR)
lineSubPathArray(1).Closed = false
lineSubPathArray(1).entireSubPath = lineArray

'line#2
Set lineArray2(1) = CreateObject("Photoshop.PathPointInfo")
lineArray2(1).Kind = 2
lineArray2(1).Anchor = Array(150, 200)
lineArray2(1).LeftDirection = lineArray2(1).Anchor
lineArray2(1).RightDirection = lineArray2(1).Anchor

Set lineArray2(2) = CreateObject("Photoshop.PathPointInfo")
lineArray2(2).Kind = 2
lineArray2(2).Anchor = Array(200, 100)
lineArray2(2).LeftDirection = lineArray2(2).Anchor
lineArray2(2).RightDirection = lineArray2(2).Anchor

Set lineSubPathArray(2) = CreateObject("Photoshop.SubPathInfo")
lineSubPathArray(2).operation = 2
lineSubPathArray(2).Closed = false
lineSubPathArray(2).entireSubPath = lineArray2
'draw the "ice cream" curve above the two lines already on the screen
'it's a curved line, so there are 3 points, not 2 and the
'coordinates for the middle point (lineArray3(1)) are different.
'The left direction is positioned "above" the anchor on the screen.
'The right direction is positioned "below" the anchor
'You can change the coordinates for these points to see
'how the curve works...
Set lineArray3(1) = CreateObject("Photoshop.PathPointInfo")
lineArray3(1).Kind = 2
lineArray3(1).Anchor = Array(200, 100)
lineArray3(1).LeftDirection = lineArray3(1).Anchor
lineArray3(1).RightDirection = lineArray3(1).Anchor

Set lineArray3(2) = CreateObject("Photoshop.PathPointInfo")
lineArray3(2).Kind = 2
lineArray3(2).Anchor = Array(150, 50)
lineArray3(2).LeftDirection = Array(100, 50)
lineArray3(2).RightDirection = Array(200, 50)

Set lineArray3(3) = CreateObject("Photoshop.PathPointInfo")
lineArray3(3).Kind = 2
lineArray3(3).Anchor = Array(100, 100)
lineArray3(3).LeftDirection = lineArray3(3).Anchor
lineArray3(3).RightDirection = lineArray3(3).Anchor

Set lineSubPathArray(3) = CreateObject("Photoshop.SubPathInfo")
lineSubPathArray(3).operation = 2
lineSubPathArray(3).Closed = false
lineSubPathArray(3).entireSubPath = lineArray3

'create the path item
Set myPathItem = docRef.PathItems.Add("A Line", lineSubPathArray)
```

```
' stroke it so we can see something
myPathItem.StrokePath(2) 'for PsToolType --> 2 (psBrush)

' Reset the application preferences
Preferences.RulerUnits = startRulerUnits
Preferences.TypeUnits = startTypeUnits
DisplayDialogs = startDisplayDialogs
```

PathItems

The collection of `PathItem` objects in the document.

Note: See 'PathItem' on page 95 for information on `PathItem` objects.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of `PathItem` objects in the `PathItems` collection.
Item	Object (PathItem)	Read-only. Gets an element from the collection.
Parent	Object (Document)	Read-only. The `PathItems` object's container.
typename	String	Read-only. The class name of the referenced `PathItems` object.

Methods

Method	Parameter Type	Returns	What it does
Add (Name, EntirePath)	String Array (SubPathItem objects)	PathItem	Creates a new `PathItem` object.
Index (ItemPtr)	Object (PathItem)	Number (Long)	Gets a `PathItem` object from the `PathItems` collection.
RemoveAll ()			Removes all `PathItem` objects from the `PathItems` collection.

PathPoint

Information about an array of `PathPointInfo` objects.

Note: You do not use the `PathPoint` object to create points that make up a path. Rather, you use the `PathPoint` object to retrieve information about the points that describe path segments. To create path points, use the `PathPointInfo` objects. See 'PathPointInfo' on page 137.

Properties

Property	Value Type	What it is
Anchor	`Array(UnitValue)`	Read-write. The point on the curve (`LeftDirection`/`RightDirection` are points representing the control handle end points).
Application	`Object (Application)`	Read-only. The application that the collection belongs to.
Kind	`PsPointKind`	Read-write. The `PathPoint` object's type.
LeftDirection	`Array(UnitValue`	Read-write. The x and y coordinates that define the left handle.
Parent	`Object (SubPathItem)`	Read-only. The `PathPoint` object's container.
RightDirection	`Array(UnitValue)`	Read-write. The x and y coordinates that define the right handle.
typename	`String`	Read-only. The class name of the referenced `PathPoint` object.

PathPointInfo

A point on a path, expressed as an array of three coordinate arrays: the anchor point, left direction point, and right direction point. For paths that are straight segments (not curved), the coordinates of all three points are the same. For curved segments, the coordinates are different. The difference between the anchor point and the left or right direction points determines the arc of the curve. You use the left direction point to bend the curve "outward" or make it convex; you use the right direction point to bend the curve "inward" or make it concave.

Properties

Property	Value Type	What it is
Anchor	Array	Read-write. The x and y coordinates of one end point of the path segment.
Application	Object (Application)	Read-only. The application that the collection belongs to.
Kind	PsPointKind	Read-write. The PathPointInfo object's kind.
LeftDirection	Array(UnitValue)	Read-write. The location of the left direction point ('in' position).
RightDirection	Array(UnitValue)	Read-write. The location of the right handle ('out' position).
typename	String	Read-only. The class name of the referenced PathPointInfo object.

PathPoints

A collection of `PathPoint` objects that comprises the `PathPoints` property of the `SubPathItem` object.

Note: See 'SubPathItem' on page 131 for more information.

Properties

Property	Value Type	What it is
`Application`	`Object (Application)`	Read-only. The application that the collection belongs to.
`Count`	`Number (Long)`	Read-only. The number of elements in the `PathPoints` collection.
`Item`	`Object (PathPoint)`	Read-only. Gets an element from the collection.
`Parent`	`Object (SubPathItem)`	Read-only. The `PathPoints` object's container.
`typename`	`String`	Read-only. The class name of the referenced `PathPoints` object.

Method	Parameter type	Returns	What it does
`Index` `(ItemPtr)`	`Object (PathPoint)`	`Number (Long)`	Gets an element from the `PathPoints` collection.

PDFOpenOptions

Options that can be specified when opening a document in generic PDF format.

Properties

Property	Value Type	What it is
AntiAlias	Boolean	Read-write. Indication of whether to use antialias.
Application	Object (Application)	Read-only. The application that the collection belongs to.
BitsPerChannel	PsBitsPerChannelType	Read-write. The number of bits per channel.
ConstrainProportions	Boolean	Deprecated for Adobe Photoshop CS2.
CropPage	PsCropToType	Read-write. The method of cropping to use.
Height	UnitValue	Deprecated for Adobe Photoshop CS2.
Mode	PsOpenDocumentMode	Read-write. The color model to use.
Name	String	Read-write. The name of the document.
Page	Number (Long)	Read-write. The page to which to open the document.
Resolution	Number (Double)	Read-write. The resolution of the document (in pixels per inch).
SuppressWarnings	Boolean	Read-write. Indication of whether to suppress warnings when opening the document.
Typename	String	Read-only. The class name of the referenced PDFOpenOptions object.
UsePageNumber	Boolean	Read-write. Indication of whether the value specified in the page property will refer to an image number when usePageNumber = false. See Page.
Width	UnitValue	Deprecated for Adobe Photoshop CS2.

PDFSaveOptions

Options that can be specified when saving a document in PDF format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels with the file.
Annotations	Boolean	Read-write. Indication of whether to save comments with the file.
Application	Object (Application)	Read-only. The application that the collection belongs to.
ColorConversion	Boolean	Read-write. Indication of whether to convert the color profile to a destination profile.
ConvertToEightBit	Boolean	Read-write. Indication of whether to convert a 16-bit image to 8-bit for better compatibility with other applications.
Descripton	String	Read-write. Description of the save options to use.
DestinationProfile	String	Read-write. Description of the final RGB or CMYK output device, such as a monitor or a press standard.
DowngradeColorProfile	Boolean	Deprecated for Adobe Photoshop CS2.
DownSample	PsPDFResampleType	Read-write. The down sample method to use.
DownSampleSize	Number (Double)	Read-write. The size to downsample images if they exceed the limit in pixels per inch.
DownSampleSizeLimit	Number (Double)	Read-write. Limits downsampling or subsampling to images that exceed this value in pixels per inch.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
EmbedFonts	Boolean	Deprecated for Adobe Photoshop CS2.
EmbedThumbnail	Boolean	Read-write. Indication of whether to include a small preview image in Adobe PDF files.
Encoding	PsPDFEncoding	Read-write. The encoding method to use. Default: 1 (psPDFZIP).
Interpolation	Boolean	Deprecated for Adobe Photoshop CS2.

Property	Value Type	What it is (Continued)
JPEGQuality	Number (Long)	Read-write. The quality of the produced image (0 - 12), which is inversely proportionate to the compression amount. **Note:** Valid only when Encoding = 2 (psPDFJPEG).
Layers	Boolean	Read-write. Indication of whether to save the document's layers.
OptimizeForWeb	Boolean	Read-write. Indication of whether to improve performance of PDF files on Web servers.
OutputCondition	String	Read-write. An optional comment field for inserting descriptions of the output condition. The text is stored in the PDF/X file.
OutputConditionID	String	Read-write. Indentifier for the output condition.
PDFCompatibility	PsPDFCompatibilityType	Read-write. The PDF version to make the document compatible with.
PDFStandard	PsPDFStandardType	Read-write. The PDF standard to make the document compatible with.
PreserveEditing	Boolean	Read-write. Indication of whether to reopen the PDF in Adobe Photoshop CS2 with native Photoshop data intact.
PresetFile	String	Read-write. The preset file to use for settings. **Note:** This option overrides other settings.
ProfileInclusionPolicy	Boolean	Read-write. Indication of whether to show which profiles to include.
RegistryName	String	Read-write. URL where the output condition is registered.
SpotColors	Boolean	Read-write. Indication of whether to save spot colors.
TileSize	Number (Long)	Read-write. Compression option. **Note:** Valid only when encoding = PDFEncoding.JPEG2000.
Transparency	Boolean	Deprecated for Adobe Photoshop CS2.

Property	Value Type	What it is (Continued)
typename	String	Read-only. The class name of the referenced PDFSaveOptions object.
UseOutlines	Boolean	Deprecated for Adobe Photoshop CS2.
VectorData	Boolean	Deprecated for Adobe Photoshop CS2.
View	Boolean	Read-write. Indication of whether to open the saved PDF in Adobe Acrobat.

PhotoCDOpenOptions

Options to be specified when opening a Kodak Photo CD (PCD) files, including high-resolution files from Pro Photo CD discs.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
ColorProfileName	String	Read-write. The profile to use when reading the image.
ColorSpace	PsPhotoCDColorSpace	Read-write. The colorspace for the image.
Orientation	PsOrientation	Read-write. The image orientation.
PixelSize	PsPhotoCDSize	Read-write. The image dimensions.
Resolution	Number (Double)	Read-write. The image resolution (in pixels per inch).
typename	String	Read-only. The class name of the referenced PhotoCDOpenOptions object.

PhotoshopSaveOptions

Options that can be specified when saving a document in PSD format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Annotations	Boolean	Read-write. Indication of whether to save the annotations.
Application	Object (Application)	Read-only. The application that the collection belongs to.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
Layers	Boolean	Read-write. Indication of whether to preserve the layers.
SpotColors	Boolean	Read-write. Indication of whether to save the spot colors.
typename	String	Read-only. The class name of the referenced PhotoshopSaveOptions object.

PICTFileSaveOptions

Options that can be specified when saving a document in PICT format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Application	Object (Application)	Read-only. The application that the collection belongs to.
Compression	PsPICTCompression	Read-write. Default: 1)
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
Resolution	PsPICTBitsPerPixels	Read-write. The number of bits per pixel.
typename	String	Read-only. The class name of the referenced PICTFileSaveOptions object.

PICTResourceSaveOptions

Options that can be specified when saving a document as a PICT Resource file.

Properties

Property	Value Type	What it is
AlphaChannelChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Application	Object (Application)	Read-only. The application that the collection belongs to.
Compression	PsPICTCompression	Read-write. The type of compression to use. Default: 1(psNoPICTCompression).
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
Name	String	Read-write. The name of the PICT resource.
Resolution	PsPICTBitsPerPixels	Read-write. The number of bits per pixel.
ResourceID	Number (Long)	Read-write. The ID of the PICT resource. Default: 128.
typename	String	Read-only. The class name of the referenced PICTResourceSaveOptions object.

PicturePackageOptions

Options that can be specified for a Picture Package.

Property	Value type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Content	PsPicturePackageTextType	Read-write. The content information. Default: 0 (psNoText).
Flatten	Boolean	Read-write. Indicates whether all layers in the final document are flattened. Default: true.
Font	PsGalleryFontType	Read-write. The font used for security text. Default: 1 (psArial).
FontSize	Number (Long)	Read-write. The font size used for security text. Default: 12.
Layout	String	Read-write. The layout to use to generate the picture package. Default: "(2)5x7".
Mode	PsNewDocumentMode	Read-write. Read-write. The color profile to use as the document mode. Default: 2 (psNewRGB).
Opacity	Number (Long)	Read-write. The Web page security opacity as a percent. Default: 100.
Resolution	Number (Double)	Read-write. The resolution of the document in pixels per inch. Default: 72.0.
Text	String	Read-write. The picture package custom text. **Note:** Valid only when Content = 2 (psUserText). See Content.
TextColor	RGBColor	Read-write. The color to use for security text.
TextPosition	PsGallerySecurityTextPositionType	Read-write. The security text position. Default: 1 (psCentered).
TextRotate	PsGallerySecurityTextRotateType	Read-write. The orientation to use for security text. Default: 1 (psZero).
typename	String	Read-only. The class name of the referenced PicturePackageOptions object.

PixarSaveOptions

Options that can be specified when saving a document in Pixar format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Application	Object (Application)	Read-only. The application that the collection belongs to.
typename	String	Read-only. The class name of the referenced PixarSaveOptions object.

PNGSaveOptions

Options that can be specified when saving a document in PNG format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Interlaced	Boolean	Read-write. Indicates whether the should rows be interlaced. Default: false.
typename	String	Read-only. The class name of the referenced PNGSaveOptions object.

Preferences

Options to define for the `Preferences` property of the `app` object. See 'Preferences' on page 17 (in the Properties table for the `app` object).

Note: Defining the `Preferences` properties is basically equivalent to selecting Edit > Preferences (Windows) or Photoshop > Preferences in the Adobe Photoshop CS2 application. For explanations of individual settings, please refer to Adobe Photoshop CS2 Help.

Properties

Property	Value Type	What it is
AdditionalPluginFolder	String	Read-write. The path to an additional plug-in folder. **Note:** Valid only when UseAdditionalPluginFolder = true. See UseAdditionalPluginFolder.
AppendExtension	PsSaveBehavior	Read-write. Save files with extensions on Windows.
Application	Object (Application)	Read-only. The application that the collection belongs to.
AskBeforeSavingLayeredTIFF	Boolean	Read-write. Indication of whether to ask the user to verify layer preservation options when saving a file in TIFF format.
AutoUpdateOpenDocuments	Boolean	Read-write. Indication of whether to automatically update open documents.
BeepWhenDone	Boolean	Read-write. Indication of whether to beep when a process finishes.
ColorChannelsInColor	Boolean	Read-write. Indication of whether to display component channels in the Channels palette in color.
ColorPicker	PsColorPicker	Read-write.
ColumnGutter	Number (Double)	Read-write. The width of the column gutters (in points). (0.1 - 600.0).
ColumnWidth	Number (Double)	Read-write. Column width (in points) (0.1 - 600.0).
CreateFirstSnapshot	Boolean	Read-write. Indication of whether to automatically make the first snapshot when a new document is created.
DynamicColorSliders	Boolean	Read-write. Indication of whether dynamic color sliders appear in the Color palette.
EditLogItems	PsEditLogItemsType	Read-write. The options for editing history log items. **Note:** Valid only when UseHistoryLog = true. See UseHistoryLog.
ExportClipboard	Boolean	Read-write. Indication of whether to retain Adobe Photoshop CS2 contents on the clipboard after you exit the application.

Property	Value Type	What it is (Continued)
FontPreviewSize	psFontPreviewType	Read-write. Indication of whether to show font previews in the type tool font menus.
GamutWarningOpacity	Number (Double)	Read-write. (0 - 100 as percent).
GridSize	PsGridSize	Read-write. The size to use for squares in the grid.
GridStyle	PsGridLineStyle	Read-write. The formatting style for non-printing grid lines.
GridSubDivisions	Number (Long)	Read-write. (1 - 100)
GuideStyle	PsGuideLineStyle	Read-write. The formatting style for non-printing guide lines.
ImageCacheForHistograms	Boolean	Read-write. Indication of whether to use the sampled data cache for histograms in the Level dialog (faster but not as accurate).
ImageCacheLevels	Number (Long)	Read-write. The number of images to hold in the cache (1 - 8).
ImagePreviews	PsSaveBehavior	Read-write. The behavior mode to use when saving files.
Interpolation	PsResampleMethod	Read-write. The method to use to assign color values to any new pixels created when an image is resampled or resized.
KeyboardZoomResizesWindows	Boolean	Read-write. Indication of whether to automatically resize the window when zooming in or out using keyboard shortcuts.
MaximizeCompatibility	PsQueryStateType	Read-write. The behavior to use to check whether to maximize compatibility when opening Adobe Photoshop CS2 (PSD) files.
MaxRAMuse	Number (Long)	Read-write. The maximum percentage of available RAM used by Adobe Photoshop CS2 (5 - 100).
NonLinearHistory	Boolean	Read-write. Indication of whether to allow non-linear history.
NumberOfHistoryStates	Number (Long)	Read-write. The number of history states to preserve (1 - 100).
OtherCursors	PsOtherPaintingCursors	Read-write. The type of pointer to use.
PaintingCursors	PsPaintingCursors	Read-write. The type of pointer to use.

Property	Value Type	What it is (Continued)
Parent	Object (Application)	Read-write. The Preferences object's container.
PixelDoubling	Boolean	Read-write. Indication of whether to halve the resolution or (double the size of pixels) to make previews display more quickly.
PointSize	PsPointType	Read-write. The point/pica size.
RecentFileListLength	Number (Long)	Read-write. The number of items in the recent file list (0 - 30).
RulerUnits	PsUnits	Read-write. The unit the scripting system will use when receiving and returning values.
SaveLogItems	PsSaveLogItemsType	Read-write. The options for saving the history items.
SaveLogItemsFile	String	Read-write. The path to the history log file.
SavePaletteLocations	Boolean	Read-write. Indication of whether to make new palette locations the default location.
ShowAsianTextOptions	Boolean	Read-write. Indication of whether to display Asian text options in the Paragraph palette.
ShowEnglishFontNames	Boolean	Read-write. Indication of whether to list Asian font names in English.
ShowSliceNumber	Boolean	Read-write. Indication of whether to display slice numbers in the document window when using the Slice tool.
ShowToolTips	Boolean	Read-write. Indication of whether to show pop up definitions on mouse over.
SmartQuotes	Boolean	Read-write. Indication of whether to use curly or straight quote marks.
typename	String	Read-only. The class name of the referenced Preferences object.
TypeUnits	PsTypeUnits	Read-write. The unit type-size that the numeric inputs are assumed to represent.
UseAdditionalPluginFolder	Boolean	Read-write. Indication of whether to use an additional folder for compatible plug-ins stored with a different application.

Property	Value Type	What it is (Continued)
UseDiffusionDither	Boolean	Read-write. Indication of whether to use diffusion dithering to minimize distinctive patterning caUsed by pattern dithering.
UseHistoryLog	Boolean	Read-write. Indication of whether to create a log file for history states.
UseLowerCaseExtension	Boolean	Read-write. Indicates whether the file extension should be lowercase.
UseShiftKeyForToolSwitch	Boolean	Read-write. Indication of whether to enable cycling through a set of hidden tools.
UseVideoAlpha	Boolean	Read-write. Indication of whether to enable Adobe Photoshop CS2 to send transparency information to your computer's video board. (Requires hardware support.)
WindowsThumbnail	Boolean	Read-write. (Requires hardware support.) Indication of whether to create a thumbnail when saving the image on Windows.

PresentationOptions

Options that can be specified for PDF presentations.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
AutoAdvance	Boolean	Read-write. Indication of whether to auto advance images when viewing the presentation. Default: true. **Note:** Valid only when Presentation = true. See Presentation.
IncludeFilename	Boolean	Read-write. Indication of whether to include the file name for the image (default: false).
Interval	Number (Long)	Read-write. The time in seconds before the view is auto advanced (1 - 60; default: 5). **Note:** Valid only when AutoAdvance = true. See AutoAdvance.
Loop	Boolean	Read-write. Indication of whether to begin the presentation again after the last page. Default: false. **Note:** Valid only when AutoAdvance = true. See AutoAdvance.
Magnification	PsMagnificationType	Read-write. The magnification type to use when viewing the image.
PDFFileOptions	PDFSaveOptions	Read-write. Options to use when creating the PDF file.
Presentation	Boolean	Read-write. Indication of whether the output will be a presentation. Default: false); when false, the output is a Multi-Page document.
Transition	PsTransitionType	Read-write. The transition from one image to the next. Default: 9 (psNoTransition). **Note:** Valid only when AutoAdvance = true. See AutoAdvance.
typename	String	Read-only. The class name of the referenced PDFPresentationOptions object.

RawFormatOpenOptions

Options that can be specified when opening a document in RAW format.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
BitsPerChannel	Number (Long)	Read-write. The number of bits for each channel. **Note:** The only valid values are BitsPerChannel = 8 or BitsPerChannel = 16.
ByteOrder	PsByteOrder	Read-write. The order in which bytes will be read. **Note:** Valid only when BitsPerChannel = 16. See BitsPerChannel.
ChannelNumber	Number (Long)	Read-write. The number of channels in the image (1 - 56). **Note:** The value of ChannelNumber cannot exceed the number of channels in the image. When BitsPerChannel = 16, only the following values are valid: 1, 3, or 4. See BitsPerChannel.
HeaderSize	Number (Long)	Read-write. The number of bytes of information that will appear in the file before actual image information begins; that is, the number of zeroes inserted at the beginning of the file as placeholders (0 - 1919999).
Height	Number (Long)	Read-write. The height of the image (in pixels).
InterleaveChannels	Boolean	Read-write. Indication of whether to store color values sequentially.
RetainHeader	Boolean	Read-write. Indication of whether to retain the header when saving. **Note:** Valid only when HeaderSize is 1 or greater.
typename	String	Read-only. The class name of the referenced RawFormatOpenOptions object.
Width	Number (Long)	Read-write. The image width in pixels.

RawSaveOptions

Options that can be specified when saving a document in RAW format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indicates whether alpha channels should be saved.
Application	Object (Application)	Read-only. The application that the collection belongs to.
SpotColors	Boolean	Read-write. Indicates whether the spot colors should be saved.
typename	String	Read-only. The class name of the referenced RawSaveOptions object.

RGBColor

The definition of a color in RGB color mode.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Blue	Number (Double)	Read-write. The blue color value (0.0 - 255.0; default: 255.0).
Green	Number (Double)	Read-write. The green color value (0.0 - 255.0; default: 255.0).
HexValue	String	Read-write. The hex representation of the color.
Red	Number (Double)	Read-write. The red color value (0.0 - 255.0; default: 255.0).
typename	String	Read-only. The class name of the referenced RGBColor object.

Selection

The selected area of a document or layer.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Bounds	array of UnitValues	Read-only. The bounding rectangle of the entire selection.
Parent	Object (Document)	Read-only. The object's container.
typename	String	Read-only. The class name of the referenced Selection object.

Methods

Method	Parameter Type	Returns	What it does
Clear ()			Clears the selection and does not copy it to the clipboard.
Contract (By)	Number (Double)		Contracts the selection by the specified amount.
Copy ([Merge])	Boolean		Copies the selection to the clipboard. When the optional argument is used and set to true, a merged copy is performed (all visible layers in the selection are copied).
Cut ()			Clears the current selection and copies it to the clipboard.
Deselect ()			Deselects the current selection.
Expand (By)	Number (Double)		Expands the selection by the specified amount.
Feather (By)	Number (Double)		Feathers the edges of the selection by the specified amount.
Fill (Filltype [, Mode] [, Opacity] [, PreserveTransparency])	Object (SolidColor, ArtLayer, HistoryState); or String PsColorBlendMode Number (Long) Boolean		Fills the selection (Opacity: 1 - 100 as percent).

Method	Parameter Type	Returns	What it does (Continued)
Grow (Tolerance, AntiAlias)	Number (Long) Boolean		Grows the selection to include all adjacent pixels falling within the specified tolerance range.
Invert ()			Inverts the selection (deselects the selection and selects the rest of the layer or document). **Note:** To flip the selection shape, see Rotate.
Load (From [, Combination] [, Inverting])	Channel PsSelectionType Boolean		Loads the selection from the specified channel.
MakeWorkPath ([Tolerance])	Number (Double)		Makes this selection item the work path for this document.
Resize ([Horizontal] [, Vertical] [, Anchor])	Number (Double) Number (Double) PsAnchorPosition		Resizes the selected area to the specified dimensions and anchor position.
ResizeBoundary ([Horizontal] [, Vertical] [, Anchor])	Number (Double) Number (Double) PsAnchorPosition		Changes the size of the selection to the specified dimensions around the specified anchor.
Rotate (Angle [, Anchor])	Number (Double) PsAnchorPosition		Rotates the selection by the specified amount around the specified anchor point.
RotateBoundary (Angle [, Anchor])	Number (Double) PsAnchorPosition		Rotates the boundary of the selection around the specified anchor.
Select (Region [, Type] [, Feather] [, AntiAlias])	Array (Points: Array (Array (x,y),...) PsSelectionType Number (Double) Boolean		Selects the specified region.
SelectAll ()			Selects the entire layer.
SelectBorder (Width)	Number (Double)		Selects the selection border only (in the specified width); subsequent actions do not affect the selected area within the borders.

Method	Parameter Type	Returns	What it does (Continued)
Similar (Tolerance, AntiAlias)	Number (Long) Boolean		Grows the selection to include pixels throughout the image falling within the tolerance range.
Smooth (Radius)	Number (Long)		Cleans up stray pixels left inside or outside a color-based selection (within the radius specified in pixels).
Store (Into [, Combination])	Channel PsSelectionType		Saves the selection as a channel.
Stroke (StrokeColor, Width [, Location] [, Mode] [, Opacity] [, PreserveTransparency])	Object (color) Number (Long) PsStrokeLocation PsColorBlendMode Number (Long) Boolean		Strokes the selection border (Opacity: 1 - 100 as percent).
Translate ([DeltaX] [, DeltaY])	UnitValue UnitValue		Moves the entire selection relative to its current position.
TranslateBoundary ([DeltaX] [, DeltaY])	UnitValue UnitValue		Moves the selection relative to its current position.

Sample Script

➤ **The following script creates a checkerboard using the following steps:**

1. Create an 800 x 800 pixel document.
2. Divide the entire document into 100 x 100 pixel squares.
3. Select every other square in the first row, then shift the selection criteria to select the alternate squares in the following row. Repeat until every other square in the document is selected.
4. Fill the selected squares with the foreground color from the palette.
5. Invert the selection and fill the newly selected squares with the background color from the palette.
6. Deselect the squares to remove the selection outlines (the “marching ants").

Selection.vbs

```
Dim appRef, startRulerUnits, startTypeUnits, startDisplayDialogs, docSize
Dim cells, cellSize, checkersDoc, shiftIt, h, v, eventWait, enumRedrawComplete
Dim typeState, keyState, desc

Set appRef = CreateObject("Photoshop.Application")

' Save the current Preferences
```

```
startRulerUnits = appRef.Preferences.RulerUnits
startTypeUnits = appRef.Preferences.TypeUnits
startDisplayDialogs = appRef.DisplayDialogs

' Set Photoshop CS2 to use pixels and display no dialogs
appRef.Preferences.RulerUnits = 1 'for PsUnits --> 1 (psPixels)
appRef.Preferences.TypeUnits = 1 'for PsTypeUnits --> 1 (psPixels)
appRef.DisplayDialogs = 3 'for PsDialogModes --> 3 (psDisplayNoDialogs)

' Close all the open documents
Do While appRef.Documents.Count
   appRef.ActiveDocument.Close
Loop

' Create variables for the 800 pixel board divided in even 100 x 100 squares
docSize = 800
cells = 8
cellSize = docSize / cells

' create a new document
Set checkersDoc = appRef.Documents.Add(docSize, docSize, 72, "Checkers")

' Create a variable to use for selecting the checker board
' That allows me to shift the selection one square to the right
'on every other row, and then shift back for the rows in between.
shiftIt = true

' loop through vertically to create the first row
v = 0
For v = 0 To (docSize - 1) Step cellSize
   ' Switch the shift For a new row
   shiftIt = Not shiftIt

   ' loop through horizontally
   h = 0
   For h = 0 To (docSize - 1) Step (cellSize * 2)
      ' push over the cellSize to start with only
      If shiftIt And h = 0 Then
         h = h + cellSize
      End If
      ' Select a square
      selRegion = Array(Array(h, v), _
                     Array(h + cellSize, v), _
                     Array(h + cellSize, v + cellSize), _
                     Array(h, v + cellSize), _
                     Array(h, v))

      If h = 0 And v = 0 Then
         checkersDoc.Selection.Select(selRegion)
      Else
         checkersDoc.Selection.Select(selRegion, 2) ' for PsSelectionType --> 2
(psExtendSelection)
      End If

      ' turn this off for faster execution
      ' turn this on for debugging
      WaitForRedraw
   Next
Next
```

```
' Fill the current selection with the Foreground color
checkersDoc.Selection.Fill(appRef.ForegroundColor)

'Invert the selection
checkersDoc.Selection.Invert()

' Fill the new selection with the background color
checkersDoc.Selection.Fill(appRef.BackgroundColor)

' Clear the selection to get rid of the non-printing borders
checkersDoc.Selection.Deselect()

' Reset the application preferences
appRef.Preferences.RulerUnits = startRulerUnits
appRef.Preferences.TypeUnits = startTypeUnits
appRef.DisplayDialogs = startDisplayDialogs



' -------------------------------------------------------------------------------
' A helper function for debugging
' It also helps the user see what is going on
' if you turn it off for this example you
' get a flashing cursor for a Number (Long) time
' Save a bit of time by creating the variables only once
Private Sub WaitForRedraw
   If VarType(eventWait) = vbEmpty Then
      eventWait = appRef.charIDToTypeID("Wait")
   End If
   If VarType(enumRedrawComplete) = vbEmpty Then
      enumRedrawComplete = appRef.charIDToTypeID("RdCm")
   End If
   If VarType(typeState) = vbEmpty Then
      typeState = appRef.charIDToTypeID("Stte")
   End If
   If VarType(keyState) = vbEmpty Then
      keyState = appRef.charIDToTypeID("Stte")
   End If
   If VarType(desc) = vbEmpty Then
      Set desc = CreateObject("Photoshop.ActionDescriptor")
      desc.putEnumerated KeyState, typeState, enumRedrawComplete
   End If
   appRef.executeAction(eventWait, desc, 3) 'for PsDialogModes --> 3
(psDisplayNoDialogs)
End Sub
```

SGIRGBSaveOptions

Options that can be specified when saving a document in SGIRGB format.

Note: The SGIRGB format is not installed automatically with Adobe Photoshop CS2.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Application	Object (Application)	Read-only. The application that the collection belongs to.
SpotColors	Boolean	Read-write. Indication of whether to save the spot colors.
typename	String	Read-only. The class name of the referenced SGIRGBSaveOptions object.

SolidColor

A color definition used in the document.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
CMYK	Object (CMYKColor)	Read-write. The CMYK color mode.
Gray	Object (GrayColor)	Read-write. The Grayscale color mode.
HSB	Object (HSBColor)	Read-write. The HSB color mode.
Lab	Object (LabColor)	Read-write. The LAB color mode.
Model	PsColorModel	Read-write. The color model.
NearestWebColor	Object (RGBColor)	Read-only. The nearest Web color to the current color.
RGB	Object (RGBColor)	Read-write. The RGB color mode.
typename	String	Read-only. The class name of the referenced SolidColor object.

Methods

Method	Parameter Type	Returns	What it does
IsEqual (Color)	SolidColor	Boolean	Indication of whether the SolidColor object is visually equal to the specified color.

SubPathInfo

An array of `PathPointInfo` objects that describes a straight or curved segment of a path.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Closed	Boolean	Read-write. Indication of whether the path describes an enclosed area.
EntireSubPath	Array (PathPoint objects)	Read-write.
Operation	PsShapeOperation	Read-write. The sub path's operation on other sub paths.
typename	String	Read-only. The class name of the referenced `SubPathInfo` object.

SubPathItem

Information about a path.

Note: You do not use the `SubPathItem` object to create a path. Rather, you use the `SubPathInfo` object to retrieve information about a path. (Note that all of the `SubPathItem` object's properties are *Read-only*.) To create path segments, see SubPathInfo.

Properties

Property	Value Type	What it is
`Application`	`Object` (`Application`)	Read-only. The application that the collection belongs to.
`Closed`	`Boolean`	Read-only. Indicates whether the path is closed.
`Operation`	`PsShapeOperation`	Read-only. The sub path operation on other sub paths.
`Parent`	`Object` (`PathItem`)	Read-only. The object's container.
`PathPoints`	`Object` (`PathPoints`)	Read-only. The `PathPoints` collection.
`typename`	`String`	Read-only. The class name of the referenced `SubPathItem` object.

SubPathItems

A collection of `SubPathItem` objects. See SubPathItem.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `SubPathItems` collection.
Item	Object (SubPathItem)	Read-only. Gets an element from the collection.
Parent	Object (PathItem)	Read-only. The `SubPathItems` object's container.
typename	String	Read-only. The class name of the referenced `SubPathItems` object.

Methods

Method	Parameter type	Returns	What it does
Index (ItemPtr)	Object (SubPathItem)	Number (Long)	Gets an element from the `SubPathItems` collection.

TargaSaveOptions

Options that can be set when saving a document in TGA (Targa) format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Application	Object (Application)	Read-only. The application that the collection belongs to.
Resolution	PsTargaBitsPerPixels	Read-write. The number of bits per pixel. Default: 24.
RLECompression	Boolean	Read-write. Indicates whether RLE compression should be used. Default: true.
typename	String	Read-only. The class name of the referenced targaSaveOptions object.

TextFont

Details about a font in the `TextFonts` collection.

Note: See TextFonts for more information on the `TextFonts` collection.

Properties

Property	Value Type	What it is
`Application`	`Object` (`Application`)	Read-only. The application that the collection belongs to.
`Family`	`String`	Read-only. The font family.
`Name`	`String`	Read-only. The name of the font.
`Parent`	`Object` (`Application`)	Read-only. The object's container.
`PostScriptName`	`String`	Read-only. The PostScript name of the font.
`Style`	`String`	Read-only. The font style.
`typename`	`String`	Read-only. The class name of the referenced `TextFont` object.

TextFonts

The collection of fonts available on your computer.

Note: The `TextFonts` object corresponds to the `fonts` property of the `app` object. In a script, you use *fonts* to refer to a `TextFonts` object. The following sample demonstrates how to use the `Count` property of the `TextFonts` object to display a dialog that indicates the number of fonts installed on the machine.

- Correct:

```
Alert appRef.Fonts.Count
```

- Incorrect:

```
Alert appRef.TextFonts.Count
```

See Application', specifically the `Fonts` property, for more information.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Count	Number (Long)	Read-only. The number of elements in the `TextFonts` collection.
Item	Object (TextFont)	Read-only. Gets an element from the collection.
Parent	Object (Application)	Read-only. The object's container.
typename	String	Read-only. The class name of the referenced `TextFonts` object.

Methods

Method	Parameter Type	Returns	What it does
Index (ItemPtr)	Object (TextFont)	Number (Long)	Gets an element from the `TextFonts` collection.

TextItem

The text in an `ArtLayer` object whose `Kind` property's value is `2`.

Note: See ArtLayer, specifically the `Kind` property, for more information.

Properties

Property	Value Type	What it is
AlternateLigatures	`Boolean`	Read-write. Indication of whether to use alternate ligatures. **Note:** Alternate ligatures are the same as Discretionary Ligatures. Please refer to Adobe Photoshop CS2 Help for more information.
AntiAliasMethod	PsAntiAlias	Read-write. The method of anti aliasing to use.
Application	`Object` (Application)	Read-only. The application that the collection belongs to.
AutoKerning	PsAutoKernType	Read-write. The auto kerning option to use.
AutoLeadingAmount	`Number (Double)`	Read-write. The percentage to use for auto. Default) leading (0.01 - 5000.00 in points). **Note:** Valid only when `UseAutoLeading = true`. See UseAutoLeading.
BaselineShift	`Number (Double)`	Read-write. The unit value to use in the baseline offset of text.
Capitalization	PsCase	Read-write. The text case.
Color	`Object` (SolidColor)	Read-write. The text color.
Contents	`String`	Read-write. The actual text in the layer.
DesiredGlyphScaling	`Number (Double)`	Read-write. The desired amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MinimumGlyphScaling and MaximumGlyphScaling.

Property	Value Type	What it is (Continued)
DesiredLetterScaling **Note:** 'Letter Scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	Number (Double)	Read-write. The amount of space between letters (100 - 500; at 0, no space is added between letters). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MinimumLetterScaling and MaximumLetterScaling.
DesiredWordScaling **Note:** 'Word Scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	Number (Double)	Read-write. The amount (percentage) of space between words (0 -1000; at 100, no additional space is added between words). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MinimumWordScaling and MaximumWordScaling.
Direction	PsDirection	Read-write. The text orientation.
FauxBold	Boolean	Read-write. Indication of whether to use faux bold. Default: `false`. **Note:** Using `FauxBold.true` is equivalent to selecting text and clicking the Faux Bold button in the Character palette.
FauxItalic	Boolean	Read-write. Indication of whether to use faux italic. Default: `false`. **Note:** Using `FauxItalic.true` is equivalent to selecting text and clicking the Faux Italic button in the Character palette.
FirstLineIndent	Number (Double)	Read-write. The amount (unit value) to indent the first line of paragraphs (-1296 - 1296).
Font	String	Read-write. The text face of the character.
HangingPunctuation	Boolean	Read-write. Indication of whether to use roman Hanging Punctuation.
Height	Number (Double)	Read-write. The height of the bounding box (unit value) for paragraph text. **Note:** Valid only when `Kind = 2 (psParagraphText)`. See Kind.

Property	Value Type	What it is (Continued)
HorizontalScale	Number (Long)	Read-write. Character scaling (horizontal) in proportion to vertical scale (0 - 1000 in percent). See VerticalScale.
HyphenateAfterFirst	Number (Long)	Read-write. The number of letters after which hyphenation in word wrap is allowed (1 - 15).
HyphenateBeforeLast	Number (Long)	Read-write. The number of letters before which hyphenation in word wrap is allowed (1 - 15).
HyphenateCapitalWords	Boolean	Read-write. Indication of whether to allow hyphenation in word wrap of capitalized words.
HyphenateWordsLongerThan	Number (Long)	Read-write. The minimum number of letters a word must have in order for hyphenation in word wrap to be allowed (2 - 25).
Hyphenation	Boolean	Read-write. Indication of whether to use hyphenation in word wrap.
HyphenationZone	Number (Double)	Read-write. The distance at the end of a line that will cause a word to break in unjustified type (0 - 720 pica).
HyphenLimit	Number (Long)	Read-write. The maximum number of consecutive lines that can end with a hyphenated word.
Justification	PsJustification	Read-write. The paragraph justification.
Kind	PsTextType	Read-write. The text-wrap type.
Language	PsLanguage	Read-write. The language to use.
Leading	Number (Double)	Read-write. The leading amount (unit value).
LeftIndent	Number (Double)	Read-write. The amount (unit value) of space to indent text from the left (-1296 - 1296).
Ligatures	Boolean	Read-write. Indication of whether to use ligatures.
MaximumGlyphScaling	Number (Double)	Read-write. The maximum amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MinimumGlyphScaling and DesiredGlyphScaling.

Property	Value Type	What it is (Continued)
MaximumLetterScaling **Note:** 'Letter Scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	Number (Double)	Read-write. The maximum amount of space to allow between letters (100 - 500; at 0, no space is added between letters). **Note:** Valid only when Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MaximumLetterScaling and DesiredLetterScaling.
MaximumWordScaling **Note:** 'Word Scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	Number (Double)	Read-write. The maximum amount (percentage) of space to allow between words (0 -1000; at 100, no additional space is added between words). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MaximumWordScaling and DesiredWordScaling.
MinimumGlyphScaling	Number (Double)	Read-write. The minimum amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MaximumGlyphScaling and DesiredGlyphScaling.
MinimumLetterScaling **Note:** 'Letter Scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	Number (Double)	Read-write. The minimum amount (percentage) of space between letters (100 - 500; at 0, no space is removed between letters). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MaximumLetterScaling and DesiredLetterScaling.

Property	Value Type	What it is (Continued)
MinimumWordScaling **Note:** 'Word Scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	Number (Double)	Read-write. The minimum amount (percentage) of space between words (0 -1000; at 100, no space is removed between words). **Note:** Valid only when `Justification = 4 (psLeftJustified); Justification = 5 (psCenterJustified); Justification = 6 (psRightJustified);` or `Justification = 7 (psFullyJustified)`. See Justification. The following values are also required: MaximumWordScaling and DesiredWordScaling.
NoBreak	Boolean	Read-write. Indication of whether to allow words to break at the end of a line. **Tip:** When enacted on large amounts of consecutive characters, `noBreak = true` can prevent word wrap and thus may prevent some text from appearing on the screen.
OldStyle	Boolean	Read-write. Indication of whether to use old style type.
Parent	Object (ArtLayer)	Read-write. The `TextItem` object's container.
Position	Array (UnitValue)	Read-write. The position of origin for the text. The array must contain two values (unit value). **Tip:** Setting the `Position` property is basically equivalent to clicking the text tool at a point in the document to create the point of origin for text.
RightIndent	Number (Double)	Read-write. The amount of space (unit value) to indent text from the right (-1296 - 1296).
Size	Number (Double)	Read-write. The font size in points.
SpaceAfter	Number (Double)	Read-write. The amount of space (unit value) to use after each paragraph (-1296 - 1296).
SpaceBefore	Number (Double)	Read-write. The amount of space (unit value) to use before each paragraph (-1296 - 1296).
StrikeThru	PsStrikeThruType	Read-write. The text strike through option to use.
TextComposer	PsTextComposer	Read-write. The composition method to use to evaluate line breaks and optimize the specified hyphenation and Justification options. **Note:** Valid only when `Kind = 2 (psParagraphText)`. See Kind.

Property	Value Type	What it is (Continued)
Tracking	Number (Double)	Read-write. The amount of uniform spacing between multiple characters (-1000 - 10000). **Note:** Tracking units are 1/1000 of an em space. The width of an em space is relative to the current type size. In a 1-point font, 1 em equals 1 point; in a 10-point font, 1 em equals 10 points. So, for example, 100 units in a 10-point font are equivalent to 1 point.
typename	String	Read-only. The class name of the referenced TextItem object.
Underline	PsUnderlineType	Read-write. The text underlining options.
UseAutoLeading	Boolean	Read-write. Indication of whether to use a font's built-in leading information.
VerticalScale	Number (Long)	Read-write. Character scaling (vertical) in proportion to horizontal scale (0 - 1000 in percent). See HorizontalScale.
WarpBend	Number (Double)	Read-write. The warp bend percentage (-100 - 100).
WarpDirection	PsDirection	Read-write. The warp direction.
WarpHorizontalDistortion	Number (Double)	Read-write. The horizontal distortion (as percentage) of the warp (-100 - 100).
WarpStyle	PsWarpStyle	Read-write. The style of warp to use.
WarpVerticalDistortion	Number (Double)	Read-write. The vertical distortion (as percentage) of the warp (-100 - 100).
Width	Number (Double)	Read-write. The width of the bounding box (unit value) for paragraph text. **Note:** Valid only when Kind = 2 (psParagraphText). See Kind.

Methods

Method	Parameter Type	Returns	What it does
ConvertToShape ()			Converts the text item and its containing layer to a fill layer with the text changed to a clipping path.
CreatePath ()			Creates a clipping path from the outlines of the actual text items (such as letters or words).

TiffSaveOptions

Options that can be specified when saving a document in TIFF format.

Properties

Property	Value Type	What it is
AlphaChannels	Boolean	Read-write. Indication of whether to save the alpha channels.
Annotations	Boolean	Read-write. Indication of whether to save the annotations.
Application	Object (Application)	Read-only. The application that the collection belongs to.
ByteOrder	PsByteOrder	Read-write. The order in which the document's bytes will be read. The default is 2 (psMacOSByteOrder) when running on Mac OS and 1 (psIBMByteOrder) when running on a PC.
EmbedColorProfile	Boolean	Read-write. Indication of whether to embed the color profile in the document.
ImageCompression	PsTIFFEncodingType	Read-write. The compression type. Default: 1 (psNoTIFFCompression).
InterleaveChannels	Boolean	Read-write. Indication of whether the channels in the image will be interleaved.
JPEGQuality	Number (Long)	Read-write. The quality of the produced image (0 - 12), which is inversely proportionate to the amount of JPEG compression. **Note:** Valid only when ImageCompression = 3 (psTiffJPEG).
LayerCompression	PsLayerCompressionType	Read-write. The method of compression to use when saving layers (as opposed to saving composite data). **Note:** Valid only when Layers = true. See Layers
Layers	Boolean	Read-write. Indication of whether to save the layers.
SaveImagePyramid	Boolean	Read-write. Indication of whether to preserve multiresolution information. Default: false.
SpotColors	Boolean	Read-write. Indication of whether to save the spot colors.

Property	Value Type	What it is (Continued)
Transparency	Boolean	Read-write. Indication of whether to save the transparency as an additional alpha channel when the file is opened in another application.
typename	String	Read-only. The class name of the referenced TIFFSaveOptions object.

XMPMetadata

Camera raw image file settings stored in an XMP file in the same folder as the raw file with the same base name and an XMP extension.

Properties

Property	Value Type	What it is
Application	Object (Application)	Read-only. The application that the collection belongs to.
Parent	Object (Document)	Read-only. The object's container.
RawData	String	Read-only. The raw XML form of file information.
typename	String	Read-only. The class name of the referenced XMPMetadata object.

3 Action Manager

Adobe Photoshop CS2 actions allow you to save time by automating repetitive tasks. You create and run actions in the application interface using the Actions palette.

You can also manage actions in scripts using a utility called the *Action Manager*. The Action Manager allows you to write scripts that target Adobe Photoshop CS2 functionality that is not otherwise accessible in the scripting interface, such as third party plug-ins and filters that are available through the application but not in the current scripting reference.

The only requirement for using the Action Manager is that the task that you want to access from the Action Manager is recordable.

This chapter describes how to use the Action Manager and the scripting interface objects it includes.

The ScriptListener Plug-In

Before you use the Action Manager, you must install the ScriptListener plug-in. ScriptListener records a log file that contains code corresponding to the actions you perform in the UI.

Tip: Because ScriptListener records most of your actions, install ScriptListener only when you are creating an Action Manager. Leaving ScriptListener installed continuously will create large log files that occupy memory on your hard drive.

When you perform a task or series of tasks in Adobe Photoshop CS2, ScriptListener creates the file `C:\ScriptingListenerVB.log`.

Installing ScriptListener

The ScriptListener plug-in is located in the `..\Adobe Photoshop CS2\Scripting Guide\Utilities` folder.

➤ **To install the ScriptListener:**

1. Select the file `ScriptListener.8li` and then choose Edit > Copy.
2. Paste the file copy to the following location:

   `..\Adobe Photoshop CS\Plug-Ins\Adobe Photoshop Only\Automate`

3. Open Adobe Photoshop CS2.

   Note: If Adobe Photoshop CS2 is already open, close it and then start it again.

➤ **To uninstall the ScriptListener:**

1. Close Adobe Photoshop CS2.
2. Verify that a copy of the file `ScriptListener.8li` still exists in the `..\Adobe Photoshop CS2\Scripting Guide\Utilities` folder.

3. Delete the file `ScriptListener.8li` from the following location:

   `..\Adobe Photoshop CS\Plug-Ins\Adobe Photoshop Only\Automate`

4. Delete the log file `ScriptingListenerVB.log` from your C:\ drive.

Note: Even though you remove the ScriptListener from the Automate folder, it may continue to record actions. To prevent the `ScriptingListenerVB.log` file from becoming too large, delete it each time you finish playing a Adobe Photoshop CS2 action.

Action Manager Scripting Objects

The objects ActionDescriptor, ActionList and ActionReference are part of the Action Manager functionality.

Using the Action Manager from a VBS Script

The section demonstrates how to create the `ScriptingListenerVB.log` log file and use its contents to create your script.

The procedures in this section uses the Action Manager to make the Emboss filter available to the scripting interface. (By default, the Emboss filter is available only via the Adobe Photoshop CS2 interface.)

Note: ScriptListener must be installed in the `Automate` folder before you begin the following procedure. See Installing ScriptListener.

➤ **To make the Emboss filter scriptable:**

1. Open Adobe Photoshop CS2, then open a document.
2. Choose Window > Actions, then choose New Action from the Actions palette menu.
3. Name the action, then click Record.
4. Choose Filter > Stylize > Emboss.
5. Using the following settings:
   - Angle: 135
   - Height: 3
   - Amount: 100

6. Open `C:\ScriptingListenerVB.log`.

   At the end of the file you will see code similar to the following (although your numbers may be different):

```
DIM objApp
SET objApp = CreateObject("Photoshop.Application")
REM Use dialog mode 3 for show no dialogs
DIM dialogMode
dialogMode = 3
DIM id9
id9 = objApp.CharIDToTypeID( "Embs" )
    DIM desc4
    SET desc4 = CreateObject( "Photoshop.ActionDescriptor" )
    DIM id10
    id10 = objApp.CharIDToTypeID( "Angl" )
    Call desc4.PutInteger( id10, 135 )
    DIM id11
    id11 = objApp.CharIDToTypeID( "Hght" )
    Call desc4.PutInteger( id11, 3 )
    DIM id12
    id12 = objApp.CharIDToTypeID( "Amnt" )
    Call desc4.PutInteger( id12, 100 )
Call objApp.ExecuteAction( id9, desc4, dialogMode )
```

Note: ScriptListener separates logged commands with horizontal lines composed of hyphens (-----...). If this is not the first action recorded in the log, you can easily locate the most recent action; it follows the final hyphen-line.

7. In the script, identify the values that you used with the filter (135, 3 and 100), then copy the VB code from `ScriptListenerVB.log` to another file and substitute the filter specification values with variable names.

   In the following example, `135` has been replaced with `angle`; `3` has been replaced with `height`; `100` has been replaced with `amount`.

```
DIM objApp
SET objApp = CreateObject("Photoshop.Application")
REM Use dialog mode 3 for show no dialogs
DIM dialogMode
dialogMode = 3
DIM id9
id9 = objApp.CharIDToTypeID( "Embs" )
    DIM desc4
    SET desc4 = CreateObject( "Photoshop.ActionDescriptor" )
    DIM id10
    id10 = objApp.CharIDToTypeID( "Angl" )
    Call desc4.PutInteger( id10, angle)
    DIM id11
    id11 = objApp.CharIDToTypeID( "Hght" )
    Call desc4.PutInteger( id11, height )
    DIM id12
    id12 = objApp.CharIDToTypeID( "Amnt" )
    Call desc4.PutInteger( id12, amount )
Call objApp.ExecuteAction( id9, desc4, dialogMode )
```

8. Wrap the code in a VBScript function. In the following example, the function name is `emboss`.

```
Function Emboss( angle, height, amount )
{
   DIM objApp
```

```
SET objApp = CreateObject("Photoshop.Application")
REM Use dialog mode 3 for show no dialogs
DIM dialogMode
dialogMode = 3
DIM id9
id9 = objApp.CharIDToTypeID( "Embs" )
    DIM desc4
    SET desc4 = CreateObject( "Photoshop.ActionDescriptor" )
    DIM id10
    id10 = objApp.CharIDToTypeID( "Angl" )
    Call desc4.PutInteger( id10, 135 )
    DIM id11
    id11 = objApp.CharIDToTypeID( "Hght" )
    Call desc4.PutInteger( id11, 3 )
    DIM id12
    id12 = objApp.CharIDToTypeID( "Amnt" )
    Call desc4.PutInteger( id12, 100 )
Call objApp.ExecuteAction( id9, desc4, dialogMode )
}
```

9. To use a VBScript to apply the Emboss filter to a document, include the emboss function in the script and call the function with the desired parameters. For example, the following example applies the Emboss filter with angle 75, height 2, and amount 89.

```
'Open the document in the script
'Call emboss with desired parameters
emboss( 75, 2, 89 );
'finish the script

'include the function in the script file
Function Emboss(angle, height, amount )
{
   DIM objApp
SET objApp = CreateObject("Photoshop.Application")
REM Use dialog mode 3 for show no dialogs
DIM dialogMode
dialogMode = 3
DIM id9
id9 = objApp.CharIDToTypeID( "Embs" )
    DIM desc4
    SET desc4 = CreateObject( "Photoshop.ActionDescriptor" )
    DIM id10
    id10 = objApp.CharIDToTypeID( "Angl" )
    Call desc4.PutInteger( id10, 135 )
    DIM id11
    id11 = objApp.CharIDToTypeID( "Hght" )
    Call desc4.PutInteger( id11, 3 )
    DIM id12
    id12 = objApp.CharIDToTypeID( "Amnt" )
    Call desc4.PutInteger( id12, 100 )
Call objApp.ExecuteAction( id9, desc4, dialogMode )
}
```

Running JavaScript based Action Manager code from VBScript

You can also access JavaScript-based Action Manager code from a VBScript using the `DoJavaScriptFile` command. See 'DoJavaScriptFile' on page 18 for more information.

➤ **To execute JavaScript-based Action Manager code from a VBScript:**

1. Save the following script in a file called "C:\Emboss.jsx"

```
function emboss( angle, height, amount )
{
   var id32 = charIDToTypeID( "Embs" );
   var desc7 = new ActionDescriptor();
   var id33 = charIDToTypeID( "Angl" );
   desc7.putInteger( id33, angle );
   var id34 = charIDToTypeID( "Hght" );
   desc7.putInteger( id34, height );
   var id35 = charIDToTypeID( "Amnt" );
   desc7.putInteger( id35, amount );
   executeAction( id32, desc7 );
}
// Call emboss with values provided in the "arguments" collection
emboss( arguments[0], arguments[1], arguments[2] );
```

2. From VBScript you can then run the Emboss filter by saying:

```
Set objApp = CreateObject("Photoshop.Application")
objApp.DoJavaScriptFile "C:\Emboss.jsx", Array(75, 2, 89)
```

4 Scripting Constants

This section lists and describes the enumerations defined for use with Adobe Photoshop CS2 VBScript properties and methods.

Constant type	Values	What it means
PsAdjustmentReference	1 (psRelative) 2 (psAbsolute)	Method to use for interpreting selective color adjustment specifications: 1 = % of the existing color amount; 2 = % of the whole.
PsAnchorPosition	1 (psTopLeft) 2 (psTopCenter) 3 (psTopRight) 4 (psMiddleLeft) 5 (psMiddleCenter) 6 (psMiddleRight) 7 (psBottomLeft) 8 (psBottomCenter) 9 (psBottomRight)	The point on the object that does not move when the object is rotated or resized.
PsAntiAlias	1 (psNoAntialias) 2 (psSharp) 3 (psCrisp) 4 (psStrong) 5 (psSmooth)	Method to use to smooth edges by softening the color transition between edge pixels and background pixels.
PsAutoKernType	1 (psManual) 2 (psMetrics) 3 (psOptical)	The type of kerning to use for characters.
PsBatchDestinationType	1 (psNoDestination) 2 (psSaveAndClose) 3 (psFolder)	The destination, if any, for batch-processed files: 1: Leave all files open; 2: Save changes and close the files; 3: Save modified versions of the files to a new location (leaving the originals unchanged).
PsBitmapConversionType	1 (psHalfThreshold) 2 (psPatternDither) 3 (psDiffusionDither) 4 (psHalftoneScreen) 5 (psCustomPattern)	Specifies the quality of an image you are converting to bitmap mode.
PsBitmapHalfToneType	1 (psHalftoneRound) 2 (psHalftoneDiamond) 3 (psHalftoneEllipse) 4 (psHalftoneLine) 5 (psHalftoneSquare) 6 (psHalftoneCross)	Specifies the shape of the dots (ink deposits) in the halftone screen.
PsBitsPerChannelType	1 (psDocument1Bit) 8 (psDocument8Bits) 16 (psDocument16Bits) 32 (psDocument32Bits)	The number of bits per color channel.

Constant type	Values	What it means
PsBlendMode	1 (psPassThrough) 2 (psNormalBlend) 3 (psDissolve) 4 (psDarken) 5 (psMultiply) 6 (psColorBurn) 7 (psLinearBurn) 8 (psLighten) 9 (psScreen) 10 (psColorDodge) 11 (psLinearDodge) 12 (psOverlay) 13 (psSoftLight) 14 (psHardLight) 15 (psVividLight) 16 (psLinearLight) 17 (psPinLight) 18 (psDifference) 19 (psExclusion) 20 (psHue) 21 (psSaturationBlend) 22 (psColorBlend) 23 (psLuminosity)	Controls how pixels in the image are blended.
PsBMPDepthType	1 (psBMP1Bit) 4 (psBMP4Bits) 8 (psBMP8Bits) 16 (psBMP16Bits) 24 (psBMP24Bits) 32 (psBMP32Bits) 60 (psBMP_X1R5G5B5) 61 (psBMP_A1R5G5B5) 62 (psBMP_R5G6B5) 63 (psBMP_X4R4G4B4) 64 (psBMP_A4R4G4B4) 65 (psBMP_R8G8B8) 66 (psBMP_X8R8G8B8) 67 (psBMP_A8R8G8B8)	The number of bits per channel (also called pixel depth or color depth). The number selected indicates the exponent of 2. For example, a pixel with a bit-depth of 8 has 2^8, or 256, possible color values.
PsByteOrder	1 (psIBMByteOrder) 2 (psMac OSByteOrder)	The order in which bytes will be read.
PsCameraRAWSettingsType	0 (psCameraDefault) 1 (psSelectedImage) 2 (psCustomSettings)	The default CameraRaw settings to use: the camera settings, custom settings, or the settings of the selected image.
PsCameraRAWSize	0 (psMinimumCameraRAW) 1 (psSmallCameraRAW) 2 (psMediumCameraRAW) 3 (psLargeCameraRAW) 4 (psExtraLargeCameraRAW) 5 (psMaximumCameraRAW)	The camera RAW size type options: 0 = 1536 x 1024 1 = 2048 x 1024 2 = 3072 x 1024 4 = 4096 x 1024 5 = 5120 x 1024
PsCase	1 (psNormalCase) 2 (psAllCaps) 3 (psSmallCaps)	The case usage for type.

Constant type	Values	What it means
PsChangeMode	1 (psConvertToGrayscale) 2 (psConvertToRGB) 3 (psConvertToCMYK) 4 (psConvertToLab) 5 (psConvertToBitmap) 6 (psConvertToIndexedColor) 7 (psConvertToMultiChannel)	The type of color mode to use. **Note:** Color images must be changed to grayscale (1) mode before you can change them to bitmap (5) mode.
PsChannelType	1 (psComponentChannel) 2 (psMaskedAreaAlphaChannel) 3 (psSelectedAreaAlphaChannel) 4 (psSpotColorChannel)	The type of channel: 1: related to document color mode; 2: Alpha channel where color indicates masked area; 3: Alpha channel where color indicates selected area; 4: channel that contains spot colors.
PsColorBlendMode	2 (psNormalBlendColor) 3 (psDissolveBlend) 4 (psDarkenBlend) 5 (psMultiplyBlend) 6 (psColorBurnBlend) 7 (psLinearBurnBlend) 8 (psLightenBlend) 9 (psScreenBlend) 10 (psColorDodgeBlend) 11 (psLinearDodgeBlend) 12 (psOverlayBlend) 13 (psSoftLightBlend) 14 (psHardLightBlend) 15 (psVividLightBlend) 16 (psLinearLightBlend) 17 (psPinLightBlend) 18 (psDifferenceBlend) 19 (psExclusionBlend) 20 (psHueBlend) 21 (psSaturationBlendColor 22 (psColorBlendMode) 23 (psLuminosityBlend 24 (psBehindBlend) 25 (psClearBlend)	Controls how pixels in the image are blended.
PsColorModel	1 (psGrayscaleModel) 2 (psRGBModel) 3 (psCMYKModel) 4 (psLabModel) 5 (psHSBModel) 50 (psNoModel)	The color model to use.
PsColorPicker	1 (psAdobeColorPicker) 2 (psAppleColorPicker) 3 (psWindowsColorPicker) 4 (psPlugInColorPicker)	The color picker to use.
PsColorProfileType	1 (psNo) 2 (psWorking) 3 (psCustom)	The color profile type to use to manage this document.

Constant type	Values	What it means
PsColorReductionType	0 (psPerceptualReduction) 1 (psSelective) 2 (psAdaptive) 3 (psRestrictive) 4 (psCustomReduction) 5 (psBlackWhiteReduction) 6 (psSFWGrayscale) 7 (psMacintoshColors) 8 (psWindowsColors)	The color reduction algorithm option to use.
PsColorSpaceType	0 (psAdobeRGB) 1 (psColorMatchRGB) 2 (psProPhotoRGB) 3 (psSRGB)	The type of color space to use.
PsCopyrightedType	1 (psCopyrightedWork) 2 (psPublicDomain) 3 (psUnmarked)	The copyright status of the document.
PsCreateFields	1 (psDuplication) 2 (psInterpolation)	The method to use for creating fields.
PsCropToType	0 (psBoundingBox) 1 (psMediaBox) 2 (psCropBox) 3 (psBleedBox) 4 (psTrimBox) 5 (psArtBox)	The style to use when cropping a page.
PsDCSType	1 (psNoComposite) 2 (psGrayscaleComposite) 3 (psColorComposite)	The DCS format to use: 1: Does not create a composite file; 2: Creates a grayscale composite file in addition to DCS files; 3: Creates a color composite file in addition to DCS files.
PsDepthMapSource	1 (psNoSource) 2 (psTransparencyChannel) 3 (psLayerMask) 4 (psImageHighlight)	What to use for the depth map.
PsDescValueType	1 (psIntegerType) 2 (psDoubleType) 3 (psUnitDoubleType) 4 (psStringType) 5 (psBooleanType) 6 (psListType) 7 (psObjectType) 8 (psEnumeratedType) 9 (psReferenceType) 10 (psClassType) 11 (psAliasType) 12 (psSmartObject)	The value type of an object.
PsDialogModes	1 (psDisplayAllDialogs) 2 (psDisplayErrorDialogs) 3 (psDisplayNoDialogs)	Controls the type (mode) of dialogs Photoshop displays when running scripts.
PsDirection	1 (psHorizontal) 2 (psVertical)	The orientation of the object.

Constant type	Values	What it means
PsDisplacementMapType	1 (psStretchToFit) 2 (psTile)	Describes how the displacement map fits the image if the image is not the same size as the map.
PsDitherType	1 (psNoDither) 2 (psDiffusion) 3 (psPattern) 4 (psNoise)	The default type of dithering to use.
PsDocumentFill	1 (psWhite) 2 (psBackgroundColor) 3 (psTransparent)	The fill of the document.
PsDocumentMode	1 (psGrayscale) 2 (psRGB) 3 (psCMYK) 4 (psLab) 5 (psBitmap) 6 (psIndexedColor) 7 (psMultiChannel) 8 (psDuotone)	The color mode of the open document.
PsEditLogItemsType	1 (psSessionOnly) 2 (psConcise) 3 (psDetailed)	The history log edit options: 1: Save history log only for the session; 2: Save a concise history log; 3: Save a detailed history log.
PsElementPlacement	0 (psPlaceInside) 1 (psPlaceAtBeginning) 2 (psPlaceAtEnd) 3 (psPlaceBefore) 4 (psPlaceAfter)	The object's position in the Layers palette. **Note:** Not all values are valid for all object types. Please refer to the object property definition in VBScript Interface to make sure you are using a valid value.
PsEliminateFields	1 (psOddFields) 2 (psEvenFields)	The type of fields to eliminate.
PsExportType	1 (psIllustratorPaths) 2 (psSaveForWeb)	The export options to use.
PsExtensionType	2 (psLowercase) 3 (psUppercase)	The formatting of the extension in the filename.

Constant type	Values	What it means
PsFileNamingType	1 (psDocumentNameMixed) 2 (psDocumentNameLower) 3 (psDocumentNameUpper) 4 (psSerialNumber1) 5 (psSerialNumber2) 6 (psSerialNumber3) 7 (psSerialNumber4) 8 (psSerialLetterLower) 9 (psSerialLetterUpper) 10 (psMmddyy) 11 (psMmdd) 12 (psYyyymmdd) 13 (psYymmdd) 14 (psYyddmm) 15 (psDdmmyy) 16 (psDdmm) 17 (psExtensionLower) 18 (psExtensionUpper)	File naming options for the batch command.
psFontPreviewType	0 (psFontPreviewNone) 1 (psFontPreviewSmall) 2 (psFontPreviewMedium) 3 (psFontPreviewLarge)	The type size to use for font previews in the type tool font menus.
PsForcedColors	1 (psNoForced) 2 (psBlackWhite) 3 (psPrimaries) 4 (psWeb)	The type of colors to be forced (included) into the color table: 2: Pure black and pure white; 3: Red, green, blue, cyan, magenta, yellow, black, and white; 4: the 216 web-safe colors.
PsFormatOptionsType	1 (psStandardBaseline) 2 (psOptimizedBaseline) 3 (psProgressive)	The option with which to save a JPEG file: 1: Format recognized by most web browsers; 2: Optimized color and a slightly reduced file size; 3: Displays a series of increasingly detailed scans as the image downloads.
PsGalleryConstrainType	1 (psConstrainWidth) 2 (psConstrainHeight) 3 (psConstrainBoth)	The type of proportions to constrain for images.
PsGalleryFontType	1 (psArial) 2 (psCourierNew) 3 (psHelvetica) 4 (psTimesNewRoman)	The fonts to use for the Web photo gallery captions and other text.
PsGallerySecurityTextPositionType	1 (psParagraphText) 2 (psUpperLeft) 3 (psLowerLeft) 4 (psUpperRight) 5 (psLowerRight)	The position of the text displayed over gallery images as an antitheft deterrent.
PsGallerySecurityTextRotateType	1 (psZero) 2 (psClockwise45) 3 (psClockwise90) 4 (psCounterClockwise45) 5 (psCounterClockwise90)	The orientation of the text displayed over gallery images as an antitheft deterrent.

Constant type	Values	What it means
PsGallerySecurityType	1 (psNoSecurity) 2 (psCustomSecurityText) 3 (psFilename) 4 (psCopyright) 5 (psCaption) 6 (psCredit) 7 (psTitle)	The content to use for text displayed over gallery images as an antitheft deterrent. **Note:** All types draw from the image's file information except 2.
PsGalleryThumbSizeType	1 (psSmall) 2 (psMedium) 3 (psLarge) 4 (psCustomThumbnail)	The size of thumbnail images in the web photo gallery.
PsGeometry	0 (psTriangle) 1 (psPentagon) 2 (psHexagon) 3 (psSquareGeometry) 4 (psHeptagon) 5 (psOctagon)	Geometric options for shapes, such as the iris shape in the Lens Blur Filter.
PsGridLineStyle	1 (psGridSolidLine) 2 (psGridDashedLine) 3 (psGridDottedLine)	The line style for the nonprinting grid displayed over images.
PsGridSize	1 (psNoGrid) 2 (psSmallGrid) 3 (psMediumGrid) 4 (psLargeGrid)	The value of grid line spacing.
PsGuideLineStyle	1 (psGuideSolidLine) 2 (psGuideDashedLine)	The line style for nonprinting guides displayed over images.
PsIllustratorPathType	1 (psDocumentBounds) 2 (psAllPaths) 3 (psNamedPath)	The paths to export.
PsIntent	1 (psPerceptual) 2 (psSaturation) 3 (psRelativeColorimetric) 4 (psAbsoluteColorimetric)	The rendering intent to use when converting from one color space to another.
PsJavaScriptExecutionMode	1 (psNeverShowDebugger) 2 (psDebuggerOnError) 3 (psBeforeRunning)	The debugging behavior to use when executing a JavaScript.
PsJustification	1 (psLeft) 2 (psCenter) 3 (psRight) 4 (psLeftJustified) 5 (psCenterJustified) 6 (psRightJustified) 7 (psFullyJustified)	The placement of paragraph text within the bounding box.

Constant type	Values	What it means
PsLanguage	1 (psEnglishUSA) 2 (psEnglishUK) 3 (psCanadianFrench) 4 (psFrench) 5 (psFinnish) 6 (psGerman) 7 (psOldGerman) 8 (psSwissGerman) 9 (psItalian) 10 (psNorwegian) 11 (psNynorskNorwegian) 12 (psPortuguese) 13 (psBrazillianPortuguese) 14 (psSpanish) 15 (psSwedish) 16 (psDutch) 17 (psDanish)	The language to use.
PsLayerCompressionType	1 (psRLELayerCompression) 2 (psZIPLayerCompression)	Compression methods for data for pixels in layers.
PsLayerKind	1 (psNormalLayer) 2 (psTextLayer) 3 (psSolidFillLayer) 4 (psGradientFillLayer) 5 (psPatternfillLayer) 6 (psLevelsLayer) 7 (psCurvesLayer) 8 (psColorBalanceLayer) 9 (psBrightnessContrastLayer) 10 (psHueSaturationLayer) 11 (psSelectiveColorLayer) 12 (psChannelMixerLayer) 13 (psGradientMapLayer) 15 (psThresholdLayer) 14 (psInversionLayer) 16 (psPosterizeLayer) 17 (psSmartObjectLayer)	The kind of ArtLayer object.
PsLayerType	1 (psArtLayer) 2 (psLayerSet)	The kind of layer object.
PsLensType	1 (psZoomLens) 2 (psPrime35) 3 (psPrime105) 5 (psMoviePrime)	The type of lens to use.
PsMagnificationType	0 (psActualSize 1 (psFitPage	The type of magnification to use when viewing an image.
PsMatteType	1 (psNoMatte) 2 (psForegroundColorMatte) 3 (psBackgroundColorMatte) 4 (psWhiteMatte) 5 (psBlackMatte) 6 (psSemiGray) 7 (psNetscapeGrayMatte)	The color to use for matting.
PsNewDocumentMode	1 (psNewGray) 2 (psNewRGB) 3 (psNewCMYK) 4 (psNewLab) 5 (psNewBitmap)	The color profile to use for the document.

Constant type	Values	What it means
PsNoiseDistribution	1 (psUniformNoise) 2 (psGaussianNoise)	Distribution method to use when applying an Add Noise filter.
PsOffsetUndefinedAreas	1 (psOffsetSetToLayerFill) 2 (psOffsetWraparound) 3 (psOffsetRepeatEdgePixels)	Method to use to fill the empty space left by offsetting a an image or selection.
PsOpenDocumentMode	1 (psOpenGray) 2 (psOpenRGB) 3 (psOpenCMYK) 4 (psOpenLab)	The color profile to use.
PsOpenDocumentType	1 (psPhotoshopOpen) 2 (psBMPOpen) 3 (psCompuServeGIFOpen) 4 (psPhotoshopEPSOpen) 5 (psFilmstripOpen) 6 (psJPEGOpen) 7 (psPCXOpen) 8 (psPhotoshopPDFOpen) 9 (psPhotoCDOpen) 10 (psPICTFileFormatOpen) 11 (psPICTResourceFormatOpen) 12 (psPixarOpen) 13 (psPNGOpen) 14 (psRawOpen) 15 (psScitexCTOpen) 16 (psTargaOpen) 17 (psTIFFOpen) 18 (psPhotoshopDCS_1Open) 19 (psPhotoshopDCS_2Open) 20 (psAcrobatTouchUpImageOpen) 21 (psPDFOpen) 22 (psEPSOpen) 23 (psEPSPICTPreviewOpen) 24 (psEPSTIFFPreviewOpen) 25 (psAliasPIXOpen) 26 (psElectricImageOpen) 27 (psPortableBitmapOpen) 28 (psWavefrontRLAOpen) 29 (psSGIRGBOpen) 30 (psSoftImageOpen) 31 (psWirelessBitmapOpen) 32 (psCameraRAWOpen)	The format in which to open a document.
PsOperatingSystem	1 (psOS2) 2 (psWindows)	The operating system.
PsOrientation	1 (psLandscape) 2 (psPortrait)	The page orientation.
PsOtherPaintingCursors	1 (psStandardOther) 2 (psPreciseOther)	The pointer for the following tools: Eraser, Pencil, Paintbrush, Healing Brush, Rubber Stamp, Pattern Stamp, Smudge, Blur, Sharpen, Dodge, Burn, Sponge.

Constant type	Values	What it means
PsPaintingCursors	1 (psStandard) 2 (psPrecise) 3 (psBrushsize)	The pointer for the following tools: Marquee, Lasso, Polygonal Lasso, Magic Wand, Crop, Slice, Patch Eyedropper, Pen, Gradient, Line, Paint Bucket, Magnetic Lasso, Magnetic Pen, Freeform Pen, Measure, Color Sampler.
PsPaletteType	1 (psExact) 2 (psMac OSPalette) 3 (psWindowsPalette 4 (psWebPalette) 5 (psUniform) 6 (psLocalPerceptual) 7 (psLocalSelective) 8 (psLocalAdaptive) 9 (psMasterPerceptual) 10 (psMasterSelective) 11 (psMasterAdaptive) 12 (psPreviousPalette)	The palette type to use.
PsPathKind	1 (psNormalPath) 2 (psClippingPath) 3 (psWorkPath) 4 (psVectorMask) 5 (psTextMask)	The type of path.
PsPDFCompatibilityType	1 (psPDF13) 2 (psPDF14) 3 (psPDF15) 4 (psPDF16)	The PDF version to make the document compatible with.
PsPDFEncoding	0 (psPDFNone) 1 (psPDFZip) 2 (psPDFJPEG) 3 (psPDFPDFZip4Bit) 4 (psPDFJPEGHIGH) 5 (psPDFJPEGMEDHIGH) 6 (psPDFJPEGMED) 7 (psPDFJPEGMEDLOW) 8 (psPDFJPEGLOW) 9 (psPDFJPEG2000High)	Encoding and compression options to use when saving a document in PDF format.
PsPDFResampleType	0 (psNoResample) 1 (psPDFAverage) 2 (psPDFSubSample) 3 (psPDFBicubic)	The down sample method to use.
PsPDFStandardType	0 (psNoStandard) 1 (psPDFX1A2001) 2 (psPDFX1A2003) 3 (psPDFX32002) 4 (psPDFX32003)	The PDF standard to make the document compatible with.
PsPhotoCDColorSpace	1 (psRGB8) 2 (psRGB16) 3 (psLab8) 4 (psLab16)	The color space to use when creating a Photo CD.

Constant type	Values	What it means
PsPhotoCDSize	1 (psMinimumPhotoCD) 2 (psSmallPhotoCD) 3 (psMediumPhotoCD) 4 (psLargePhotoCD) 5 (psExtralargePhotoCD) 6 (psMaximumPhotoCD)	The pixel dimensions of the image.
PsPICTBitsPerPixels	2 (psPICTTwoBits) 4 (psPICTFourBits) 8 (psPICTEightBits) 16 (psPICTSixteenBits) 32 (psPICTThirtyTwoBits)	The number of bits per pixel to use when compression a PICT file. **Note:** Use 16 or 32 for RGB images; use 2, 4, or 8 for bitmap and grayscale images.
PsPICTCompression	1 (psNoPICTCompression) 2 (psJPEGLowPICT) 4 (psJPEGMediumPICT) 5 (psJPEGHighPICT) 6 (psJPEGMaximumPICT)	The type of compression to use when saving an image as a PICT file.
PsPicturePackageTextType	1 (psNoText) 2 (psUserText) 3 (psFilenameText) 4 (psCopyrightText) 5 (psCaptionText) 6 (psCreditText) 7 (psOriginText)	The function or meaning of text in a Picture Package.
PsPointKind	1 (psSmoothPoint) 2 (psCornerPoint)	The role a PathPoint plays in a PathItem.
PsPointType	1 (psPostScriptPoints) 2 (psTraditionalPoints)	The kind of measurement to use for type points: 1 = 72 points/inch; 2 = 72.27 points/inch.
PsPolarConversionType	1 (psRectangularToPolar) 2 (psPolarToRectangular)	The method of polar distortion to use.
PsPreviewType	1 (psNoPreview) 2 (psMonochromeTIFF) 3 (psEightbitTIFF)	The type of image to use as a low-resolution preview in the destination application.
PsPrintEncoding	1 (psBinaryEncoding) 2 (psJPEGEncoding) 3 (psAsciiEncoding)	The type of encoding to use.
PsPurgeTarget	1 (psUndoCaches) 2 (psHistoryCaches) 3 (psClipboardCache) 4 (psAllCaches)	Cache to be targeted in a purge operation.
PsQueryStateType	1 (psAlways) 2 (psAsk) 3 (psNever)	Permission state for queries.
PsRadialBlurMethod	1 (psSpin) 2 (psZoom)	The blur method to use.
PsRadialBlurQuality	1 (psRadialBlurDraft) 2 (psRadialBlurGood) 3 (psRadialBlurBest)	The smoothness or graininess of the blurred image.

Constant type	Values	What it means
PsRasterizeType	1 (psTextContents) 2 (psShape) 3 (psFillContent) 4 (psLayerClippingPath) 5 (psEntireLayer) 6 (psLinkedLayers)	The layer element to rasterize.
PsReferenceFormType	1 (psReferenceNameType) 2 (psReferenceIndexType) 3 (psReferenceIdentifierType) 4 (psReferenceOffsetType) 5 (psReferenceEnumeratedType) 6 (psReferencePropertyType) 7 (psReferenceClassType)	The type of an ActionReference object.
PsResampleMethod	1 (psNoResampling) 2 (psNearestNeighbor) 3 (psBilinear) 4 (psBicubic) 5 (psBicubicSharper) 6 (psBicubicSmoother)	The method to use for image interpolation.
PsRippleSize	1 (psSmallRipple) 2 (psMediumRipple) 3 (psLargeRipple)	The undulation size to use.
PsSaveBehavior	1 (psNeverSave) 2 (psAlwaysSave) 3 (psAskWhenSaving)	The application's behavior when a `Save` method is called.
PsSaveDocumentType	1 (psPhotoshopSave) 2 (psBMPSave) 3 (psCompuServeGIFSave) 4 (psPhotoshopEPSSave) 6 (psJPEGSave) 7 (psPCXSave) 8 (psPhotoshopPDFSave) 10 (psPICTFileFormatSave) 11 (psPICTResourceFormatSave 12 (psPixarSave) 13 (psPNGSave) 14 (psRawSave) 15 (psScitexCTSave) 16 (psTargaSave) 17 (psTIFFSave) 18 (psPhotoshopDCS_1Save) 19 (psPhotoshopDCS_2Save) 25 (psAliasPIXSave) 26 (psElectricImageSave) 27 (psPortableBitmapSave) 28 (psWavefrontRLASave) 29 (psSGIRGBSave) 30 (psSoftImageSave) 31 (psWirelessBitmapSave)1	The format in which to save a document.
PsSaveEncoding	1 (psBinary) 2 (psJPEGLow) 3 (psAscii) 4 (psJPEGMedium) 5 (psJPEGHigh) 6 (psJPEGMaximum)	The type of encoding to use when saving a file.

Constant type	Values	What it means
PsSaveLogItemsType	1 (psMetadata) 2 (psLogFile) 3 (psLogFileAndMetadata)	The location of history log data.
PsSaveOptions	1 (psSaveChanges) 2 (psDoNotSaveChanges) 3 (psPromptToSaveChanges)	The save options to use when the `Close` method is called to close a document.
PsSelectionType	1 (psReplaceSelection) 2 (psExtendSelection) 3 (psDiminishSelection) 4 (psIntersectSelection)	The selection behavior when a selection already exists: `1`: Replace the selected area; `2`: Add the selection to an already selected area; `3`: Remove the selection from the already selected area; `4`: Make the selection only the area where the new selection intersects the already selected area.
PsShapeOperation	1 (psShapeAdd) 2 (psShapeXOR) 3 (psShapeIntersect) 4 (psShapeSubtract)	A `subPathItem` object's behavior when it intersects another `subPathItem` object.
PsSmartBlurMode	1 (psSmartBlurNormal) 2 (psSmartBlurEdgeOnly) 3 (psSmartBlurOverlayEdge)	The method to use for smart blurring: `1` : Apply blur to entire image; `2, 3` : Apply blur only to edges of color transitions.
PsSmartBlurQuality	1 (psSmartBlurLow) 2 (psSmartBlurMedium) 3 (psSmartBlurHigh)	The blur quality to use.
PsSourceSpaceType	1 (psDocumentSpace) 2 (psProofSpace)	
PsSpherizeMode	1 (psNormalSpherize) 2 (psHorizontalSpherize) 3 (psVerticalSpherize)	The curve (or stretch shape) to use for the distortion.
PsStrikeThruType	1 (psStrikeOff) 2 (psStrikeHeight) 3 (psStrikeBox)	The style of strikethrough to use.
PsStrokeLocation	1 (psInsideStroke) 2 (psCenterStroke) 3 (psOutsideStroke)	The placement of path or selection boundary strokes.
PsTargaBitsPerPixels	16 (psTarga16Bits) 24 (psTarga24Bits) 32 (psTarga32Bits)	The resolution to use when saving an image in Targa format.
PsTextComposer	1 (psAdobeSingleLine) 2 (psAdobeEveryLine)	The composition method to use to optimize the specified hyphenation and justification options.
PsTextType	1 (psPointText) 2 (psParagraphText)	The type of text: `1`: Text that does not wrap; `2`: Text that wraps within a bounding box.

Constant type	Values	What it means
PsTextureType	1 (psBlocksTexture) 2 (psCanvasTexture) 3 (psFrostedTexture) 4 (psTinyLensTexture) 5 (psTextureFile)	The type of texture or glass surface image to load for a texturizer or glass filter.
PsTIFFEncodingType	1 (psNoTIFFCompression) 2 (psTiffLZW) 3 (psTiffJPEG) 4 (psTiffZIP)	The encoding to use for TIFF files.
PsToolType	1 (psPencil) 2 (psBrush) 3 (psEraser) 4 (psBackgroundEraser) 5 (psCloneStamp) 6 (psPatternStamp) 7 (psHealingBrush) 8 (psHistoryBrush) 9 (psArtHistoryBrush) 10 (psSmudge) 11 (psBlur) 12 (psSharpen) 13 (psDodge) 14 (psBurn) 15 (psSponge) 16 (psColorReplacementTool)	The tool selection.
PsTransitionType	1 (psBlindsHorizontal) 2 (psBlindsVertical) 3 (psDissolveTransition) 4 (psBoxIn) 5 (psBoxOut) 6 (psGlitterDown) 7 (psGlitterRight) 8 (psGlitterRightDown) 9 (psNoTransition) 10 (psRandom) 11 (psSplitHorizontalIn) 12 (psSplitHorizontalOut) 13 (psSplitVerticalIn) 14 (psSplitVerticalOut) 15 (psWipeDown) 16 (psWipeLleft) 17 (psWipeRight) 18 (psWipeUp)	The method to use to transition from one image to the next in a PDF presentation.
PsTrimType	0 (psTransparentPixels) 1 (psTopLeftPixel) 9 (psBottomRightPixel	Type of pixels to trim around an image: 9 = bottom right pixel color; 1 = top left pixel color.
PsTypeUnits	1 (psTypePixels) 4 (psTypeMM) 5 (psTypePoints)	The unit to use for measuring text characters.
PsUndefinedAreas	1 (psWrapAround) 2 (psRepeatedgePixels)	The method to use to treat undistorted areas or areas left blank in an image to which the a filter in the Distort category has been applied.

Constant type	Values	What it means
PsUnderlineType	1 (psUnderlineOff) 2 (psUnderlineRight) 3 (psUnderlineLeft)	The placement of text underlining. **Note:** 3 and 2 are valid only when PsDirection = 2.
PsUnits	1 (psPixels) 2 (psInches) 3 (psCM) 4 (psMM) 5 (psPoints) 6 (psPicas) 7 (psPercent)	The measurement unit for type and ruler increments.
PsUrgency	0 (psNone) 1 (psLow) 2 (psTwo) 3 (psThree) 4 (psFour) 5 (psNormal) 6 (psSix) 7 (psSeven) 8 (psHigh)	The editorial urgency of the artwork.
PsWarpStyle	1 (psNone) 2 (psArc) 3 (psArcLower) 4 (psArcUpper) 5 (psArch) 6 (psBulge) 7 (psShellLower) 8 (psShellUpper) 9 (psFlag) 10 (psWave) 11 (psFish) 12 (psRise) 13 (psFishEye) 14 (psInflate) 15 (psSqueeze) 16 (psTwist)	The warp style to use.
PsWaveType	1 (psSine) 2 (psTriangular) 3 (psSquare)	The type of wave to use.
PsWhiteBalanceType	0 (psAsShot) 1 (psAuto) 2 (psDaylight) 3 (psCloudy) 4 (psShade) 5 (psTungsten) 6 (psFluorescent) 7 (psFlash) 8 (psCustomCameraSettings)	The lighting conditions to use (affects color balance).
PsZigZagType	1 (psAroundCenter) 2 (psOutFromCenter) 3 (psPondRipples)	The method of zigzagging to use.

Appendix A: Event ID Codes

The following table lists events and their four-character ID codes or string identifiers for use with the `Notifier` object.

Note: **Do not include single quotes (') with four-character IDs in your code.** The single quotes are used in this table to illustrate the placement of required spaces in codes that do not contain four letters. However, string identifiers, which are longer than four characters, require double quotes in the code.

Event	4-char ID or String
3DTransform	'TdT '
Average	'Avrg'
ApplyStyle	'ASty'
Assert	'Asrt'
AccentedEdges	'AccE'
Add	'Add '
AddNoise	'AdNs'
AddTo	'AddT'
Align	'Algn'
All	'All '
AngledStrokes	'AngS'
ApplyImage	'AppI'
BasRelief	'BsRl'
Batch	'Btch'
BatchFromDroplet	'BtcF'
Blur	'Blr '
BlurMore	'BlrM'
Border	'Brdr'
Brightness	'BrgC'
CanvasSize	'CnvS'
ChalkCharcoal	'ChlC'
ChannelMixer	'ChnM'
Charcoal	'Chrc'
Chrome	'Chrm'
Clear	'Cler'

Event	4-char ID or String
Close	'Cls '
Clouds	'Clds'
ColorBalance	'ClrB'
ColorHalftone	'ClrH'
ColorRange	'ClrR'
ColoredPencil	'ClrP'
ContactSheet	"0B71D221-F8CE-11d2-B21B-0008C75B322C"
ConteCrayon	'CntC'
Contract	'Cntc'
ConvertMode	'CnvM'
Copy	'copy'
CopyEffects	'CpFX'
CopyMerged	'CpyM'
CopyToLayer	'CpTL'
Craquelure	'Crql'
CreateDroplet	'CrtD'
Crop	'Crop'
Crosshatch	'Crsh'
Crystallize	'Crst'
Curves	'Crvs'
Custom	'Cstm'
Cut	'cut '
CutToLayer	'CtTL'
Cutout	'Ct '
DarkStrokes	'DrkS'
DeInterlace	'Dntr'
DefinePattern	'DfnP'
Defringe	'Dfrg'
Delete	'Dlt '
Desaturate	'Dstt'
Deselect	'Dslc'
Despeckle	'Dspc'
DifferenceClouds	'DfrC'

Event	4-char ID or String
Diffuse	'Dfs '
DiffuseGlow	'DfsG'
DisableLayerFX	'dlfx'
Displace	'Dspl'
Distribute	'Dstr'
Draw	'Draw'
DryBrush	'DryB'
Duplicate	'Dplc'
DustAndScratches	'DstS'
Emboss	'Embs'
Equalize	'Eqlz'
Exchange	'Exch'
Expand	'Expn'
Export	'Expr'
Jumpto	'Jpto'
ExportTransparentImage	"02879e00-cb66-11d1-bc43-0060b0a13dc4"
Extrude	'Extr'
Facet	'Fct '
Fade	'Fade'
Feather	'Fthr'
Fibers	'Fbrs'
Fill	'Fl '
FilmGrain	'FlmG'
Filter	'Fltr'
FindEdges	'FndE'
FitImage	"3caa3434-cb67-11d1-bc43-0060b0a13dc4"
FlattenImage	'FltI'
Flip	'Flip'
Fragment	'Frgm'
Fresco	'Frsc'
GaussianBlur	'GsnB'
Get	'getd'
Glass	'Gls '

Event	4-char ID or String
GlowingEdges	'GlwE'
Gradient	'Grdn'
GradientMap	'GrMp'
Grain	'Grn '
GraphicPen	'GraP'
Group	'GrpL'
Grow	'Grow'
HalftoneScreen	'HlfS'
Hide	'Hd '
HighPass	'HghP'
HSBHSL	'HsbP'
HueSaturation	'HStr'
ImageSize	'ImgS'
Import	'Impr'
InkOutlines	'InkO'
Intersect	'Intr'
IntersectWith	'IntW'
Inverse	'Invs'
Invert	'Invr'
LensFlare	'LnsF'
Levels	'Lvls'
LightingEffects	'LghE'
Link	'Lnk '
Make	'Mk '
Maximum	'Mxm '
Median	'Mdn '
MergeLayers	'Mrg2'
MergeLayersOld	'MrgL'
MergeSpotChannel	'MSpt'
MergeVisible	'MrgV'
Mezzotint	'Mztn'
Minimum	'Mnm '
ModeChange	"8cba8cd6-cb66-11d1-bc43-0060b0a13dc4"

Event	4-char ID or String
Mosaic	'Msc '
Mosaic_PLUGIN	'MscT'
MotionBlur	'MtnB'
Move	'move'
NTSCColors	'NTSC'
NeonGlow	'NGlw'
Next	'Nxt '
NotePaper	'NtPr'
Notify	'Ntfy'
Null	typeNull
OceanRipple	'OcnR'
Offset	'Ofst'
Open	'Opn '
Paint	'Pnt '
PaintDaubs	'PntD'
PaletteKnife	'PltK'
Paste	'past'
PasteEffects	'PaFX'
PasteInto	'PstI'
PasteOutside	'PstO'
Patchwork	'Ptch'
Photocopy	'Phtc'
PicturePackage	"4C1ABF40-DD82-11d2-B20F-0008C75B322C"
Pinch	'Pnch'
Place	'Plc '
Plaster	'Plst'
PlasticWrap	'PlsW'
Play	'Ply '
Pointillize	'Pntl'
Polar	'Plr '
PosterEdges	'PstE'
Posterize	'Pstr'
Previous	'Prvs'

Event	4-char ID or String
Print	'Prnt'
ProfileToProfile	'PrfT'
Purge	'Prge'
Quit	'quit'
RadialBlur	'RdlB'
Rasterize	'Rstr'
RasterizeTypeSheet	'RstT'
RemoveBlackMatte	'RmvB'
RemoveLayerMask	'RmvL'
RemoveWhiteMatte	'RmvW'
Rename	'Rnm '
ReplaceColor	'RplC'
Reset	'Rset'
ResizeImage	"1333cf0c-cb67-11d1-bc43-0060b0a13dc4"
Reticulation	'Rtcl'
Revert	'Rvrt'
Ripple	'Rple'
Rotate	'Rtte'
RoughPastels	'RghP'
Save	'save'
Select	'slct'
SelectiveColor	'SlcC'
Set	'setd'
SharpenEdges	'ShrE'
Sharpen	'Shrp'
SharpenMore	'ShrM'
Shear	'Shr '
Show	'Shw '
Similar	'Smlr'
SmartBlur	'SmrB'
Smooth	'Smth'
SmudgeStick	'SmdS'
Solarize	'Slrz'

Event	4-char ID or String
Spatter	'Spt '
Spherize	'Sphr'
SplitChannels	'SplC'
Sponge	'Spng'
SprayedStrokes	'SprS'
StainedGlass	'StnG'
Stamp	'Stmp'
Stop	'Stop'
Stroke	'Strk'
Subtract	'Sbtr'
SubtractFrom	'SbtF'
Sumie	'Smie'
TakeMergedSnapshot	'TkMr'
TakeSnapshot	'TkSn'
TextureFill	'TxtF'
Texturizer	'Txtz'
Threshold	'Thrs'
Tiles	'Tls '
TornEdges	'TrnE'
TraceContour	'TrcC'
Transform	'Trnf'
Trap	'Trap'
Twirl	'Twrl'
Underpainting	'Undr'
Undo	'undo'
Ungroup	'Ungr'
Unlink	'Unlk'
UnsharpMask	'UnsM'
Variations	'Vrtn'
Wait	'Wait'
WaterPaper	'WtrP'
Watercolor	'Wtrc'
Wave	'Wave'

Event	4-char ID or String
Wind	'Wnd '
ZigZag	'ZgZg'
BackLight	'BacL'
FillFlash	'FilE'
ColorCast	'ColE'

Index

D

E

M

N

O

P

Q

R

S

T

U

V

W

X

Z