

AppleScript Scripting Reference



Adobe® Photoshop® CS2



© 2005 Adobe Systems Incorporated. All rights reserved.

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

NOTICE: All information contained herein is the property of Adobe Systems Incorporated. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Adobe Systems Incorporated. The software described in this document is furnished under license and may only be used or copied in accordance with the terms of such license.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third party rights.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, Illustrator, and Photoshop are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple, Mac, and Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. Microsoft, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. JavaScript and all Java-related marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark of The Open Group.

All other trademarks are the property of their respective owners.

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

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

Contents

1	Introduction	8
	Changes Since Earlier Versions.....	8
2	Action Manager	9
	Running JavaScript based Action Manager code from AppleScript.....	9
3	AppleScript Objects.....	10
	add noise.....	11
	adjustment options.....	12
	application.....	13
	art layer, art layers.....	15
	automatic contrast.....	18
	automatic levels.....	19
	batch options.....	20
	Bitmap mode options.....	22
	blur.....	23
	blur more.....	24
	BMP save options.....	25
	brightness and contrast.....	27
	camera raw open options.....	28
	change mode options.....	30
	channel, channels.....	31
	clouds.....	33
	CMYK color.....	34
	color balance.....	35
	color value.....	36
	contact sheet options.....	37
	curves.....	39
	custom filter.....	40
	deinterlace.....	41
	desaturate.....	42
	despeckle.....	43
	difference clouds.....	44
	diffuse glow.....	45
	displace filter.....	46
	document, documents.....	47
	dust and scratches.....	51
	EPS open options.....	52
	EPS save options.....	53
	equalize.....	55
	export options save for web.....	56
	filter options.....	58
	font, fonts.....	59
	gallery banner options.....	60
	gallery color options.....	61
	gallery images options.....	62

gallery options.....	63
gallery security options.....	65
gallery thumbnail options.....	66
Gaussian blur.....	67
GIF save options.....	68
glass filter.....	70
gray color.....	71
high pass.....	72
history state, history states.....	73
HSB color.....	74
Illustrator paths export options.....	75
indexed mode options.....	76
info-object.....	78
inversion.....	80
JPEG save options.....	81
Lab color.....	82
layer, layers.....	83
layer comp, layer comps.....	85
layer set, layer sets.....	86
lens blur.....	89
lens flare.....	91
levels adjustment.....	92
maximum filter.....	93
median noise.....	94
minimum filter.....	95
mix channels.....	96
motion blur.....	97
no color.....	98
notifier, notifiers.....	99
NTSC colors.....	100
ocean ripple.....	101
offset filter.....	102
open options.....	103
path item, path items.....	104
path point, path points.....	106
path point info.....	107
PDF open options.....	108
PDF save options.....	110
Photo CD open options.....	113
Photoshop DCS 1.0 save options.....	114
Photoshop DCS 2.0 save options.....	116
Photoshop save options.....	118
PICT file save options.....	119
PICT resource save options.....	120
picture package options.....	121
pinch.....	123
Pixar save options.....	124
PNG save options.....	125
polar coordinates.....	126
posterize.....	127
presentation options.....	128

radial blur	130
raw format open options.....	131
raw save options.....	132
RGB color	133
RGB hex color.....	134
ripple	135
save options	136
selection-object	137
selective color	138
settings-object	139
SGL RGB save options	143
sharpen	144
sharpen edges	145
sharpen more.....	146
shear.....	147
smart blur	148
spherize.....	149
sub path info	150
sub path item, sub path items	151
Targa save options	152
text-object.....	153
texture fill	161
threshold adjustment	162
TIFF save options.....	163
twirl.....	165
unsharp mask.....	166
wave filter.....	167
XMP metadata	168
zigzag.....	169
4 AppleScript Commands	170
Overview.....	170
activate.....	171
adjust	172
apply	173
apply layer style	174
change mode.....	175
close	176
clear	177
contract.....	178
convert	179
convert color.....	179
convert to shape.....	179
copy.....	180
copy merged.....	180
count.....	181
create PDF presentation	182
create photo gallery	182
create selection.....	182
create work path.....	183
crop.....	184

cut	185
delete	186
deselect	187
do action	188
do javascript	189
duplicate	190
equal colors	191
exists	192
expand	193
export	194
feather	195
fill	196
fill path	196
filter	198
flatten	199
flip canvas	200
grow	201
import annotations	202
index	203
invert	204
link	205
load	206
make	207
make clipping path	207
make work path	208
merge	209
merge visible layers	209
move	210
open	211
paste	212
print	213
purge	214
quit	215
rasterize	216
recapture	217
reset from comp	218
resize canvas	219
resize image	219
reveal all	220
rotate	221
rotate boundary	221
rotate canvas	221
save	223
scale	225
scale boundary	225
select	226
select all	226
select border	226
similar	227
smooth	228
split channels	229

store.....	230
stroke	231
stroke path.....	231
translate	232
translate boundary	233
trap	234
trim	235
unlink	236
web safe color.....	237
Appendix A: Event ID Codes	238
Index	246

This reference describes the objects and commands in the Adobe® Photoshop® CS2 AppleScript 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 AppleScript-specific features.

This book contains the following sections:

- This introduction, which describes scripting support in Adobe Photoshop CS2, and lists changes to the AppleScript interface since the previous release.
- [AppleScript Objects](#), which provides a complete reference for all Photoshop DOM objects and commands.
- [AppleScript Commands](#), which describes how to use ScriptUI, an ExtendScript component that provides a user-interface model to scripters.

Changes Since Earlier Versions

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

- Support for interapplication communication among Creative Suite 2 applications through exported ExtendScript functions and interapplication messaging. For details, see the *Creative Suite 2 Bridge JavaScript Reference*.
- The following classes have been added to the AppleScript interface:
 - `camera raw open options`, which you use to specify options when opening a document in camera raw format.
 - `export options save for web`, which you use to optimize documents for the Web.
 - `contact sheet options`, which you use to create and format contact sheets.
 - `batch options`, which you use to specify options for the Batch command.
 - `lens blur options`, which you use to specify options when applying the Lens Blur filter to a layer.

Adobe Photoshop CS2 actions allow you to save time by automating repetitive tasks.

Note: You create and run actions in the application interface using the Actions palette.

You can manage actions in scripts using a utility called the *Action Manager*. The Action Manager also 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.

Running JavaScript based Action Manager code from AppleScript

There is no Action Manager functionality in AppleScript. However, you can execute JavaScript code and files from AppleScript using the `do javascript` command. See ['do javascript' on page 189](#).

You need to rewrite your JavaScript code slightly to work with the `do javascript` command's arguments in order to get access to the AppleScript values from JavaScript. For example save the following JavaScript as `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);
}
```

In your AppleScript, call the `Emboss` function with values provided in the arguments collection, as follows:

```
emboss(arguments[0], arguments[1], arguments[2]);
```

The following AppleScript code sample runs the Emboss filter:

```
tell application "Adobe Photoshop CS2"
    do javascript (file <path to Emboss.jsx>) ↵
        with arguments { 75,2,89 }
end tell
```

Note: For information on creating JavaScripts to use with the Action Manager, see *Action Manager* chapter in the *Adobe Photoshop CS2 JavaScript Scripting Reference*, which is available on the installation CD.

3

AppleScript Objects

The objects of the AppleScript dictionary for Adobe Photoshop CS2 are presented alphabetically and in tabular format in this chapter. Where appropriate, the object class has two tables:

- **Element** — Lists the classes that inherit from the current class, as well as the methods by which the classes can be referred.
- **Property** — The properties you can use with the object, as well as each property's value type, input status, and description.

The commands that can be used with an object are listed below the object's Properties table.

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

The following sample tables and command list illustrate the format for this chapter.

Elements

Element	Refer to by
channel	name, numeric index, as a range of elements, before after another element, satisfying a test

Properties

Property	Value Type	What it is
bits per channel	Valid values: eight one sixteen	The number of bits per channel.
component channels	list of channels	Read-only. A list of the component color channels.
current channels	list of channel	The selected channels.
current layer	layer	The selected layer.
info	info-object	Read-only. The document metadata.

Valid Commands

[change mode](#)
[close](#)
[convert](#)
[count](#)

add noise

Options that may be supplied when applying an Add Noise filter.

The `add noise` class supplies properties for the [filter options](#) class. The `add noise` class is used only to define a record whose properties specify options when applying an Add Noise filter. `add noise` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create an `add noise` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>amount</code>	real	(amount: 0.1% - 400%)
<code>distribution</code>	Valid values: Gaussian uniform	
<code>monochromatic</code>	boolean	

The following code sample demonstrates how to apply the add noise filter using the `filter` command of the `add noise` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using add noise filter -
  with options {class:add noise, distribution:uniform -
    monochromatic:false}
```

adjustment options

Options used with the `adjust` method. The following classes inherit properties from the adjustment options class: [automatic contrast](#), [automatic levels](#), [brightness and contrast](#), [color balance](#), [curves](#), [desaturate](#), [equalize](#), [inversion](#), [levels adjustment](#), [mix channels](#), [posterize](#), [selective color](#), [threshold adjustment](#).

Valid Commands

[adjust](#)

application

The Adobe Photoshop CS2 application object, which contains all other Adobe Photoshop CS2 objects.

Elements

Elements	Refer to by
document	name, numeric index, as a range of elements, before/after another element, satisfying a test
font	name, numeric index, as a range of elements, before/after another element, satisfying a test

Properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
properties	record	All of this object's properties returned in a single record.
background color	color value	The color for the document's background color.
color settings	string or color object	The name of selected color setting's set.
current document	document	The frontmost document.
display dialogs	Valid Values: always error dialogs never	Controls whether and dialogs are displayed while the script is running.
foreground color	color value	The default foreground color (used to fill and stroke selections).
free memory	real	Read-only. The amount of unused memory available to Adobe Adobe Photoshop CS2.
frontmost	boolean	Read-only. Indication of whether Adobe Photoshop CS2 is the frontmost application.
locale	string	Read-only. The language location of the application.
name	Unicode text	Read-only. The application's name.
preferences folder	alias	Read-only. The full path to the Preferences folder.
scripting version	Unicode text	Read-only. The scripting interface version.

Property	Value type	What it is (Continued)
settings	settings-object	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).
version	Unicode text	Read-only. The Adobe Photoshop version.

The the first line of the following code sample directs Adobe Photoshop CS2 to carry out the script's commands. The second line uses the `activate` command to make Adobe Photoshop CS2 the frontmost application.

```
tell application "Adobe Photoshop CS2"  
  activate  
end tell
```

Valid Commands

[activate](#)
[quit](#)

art layer, art layers

Any layer that can contain data. The `art layer` class inherits properties from the [layer, layers](#) class.

Properties

Property	Value Type	What it is
<code>best type</code>	type class	Read-only. (Inherited from the <code>layer</code> class) The best type for the object's value.
<code>class type</code>	class	Read-only. (Inherited from the <code>layer</code> class) The object's class.
<code>default type</code>	type class	Read-only. (Inherited from the <code>layer</code> class) The default type for the object's value.
<code>index</code>	integer	Read-only. (Inherited from the <code>layer</code> class) The index of this instance of the object.
<code>properties</code>	record	Read-only. (Inherited from the <code>layer</code> class) All of this object's properties returned in a single record.
<code>all locked</code>	boolean	Read-only. (Inherited from the <code>layer</code> class) Indication of whether all of the layer's contents are locked or editable.
<code>blend mode</code>	Valid Values: color blend color burn color dodge darken difference dissolve exclusion hard light hue blend lighten linear burn linear dodge linear light luminosity multiply normal overlay pass through pin light saturation blend screen soft light vivid light	(Inherited from the <code>layer</code> class) The mode used when compositing an object.
<code>bounds</code>	list	Read-only. (Inherited from the <code>layer</code> class) A list of coordinates that describes the bounding rectangle of the layer.
<code>container</code>	reference	Read-only. (Inherited from the <code>layer</code> class) The object's container.

Property	Value Type	What it is (Continued)
linked layers	list of layers	Read-only. (Inherited from the <code>layer</code> class) The layers linked to this layer.
name	Unicode text	(Inherited from the <code>layer</code> class) The layer's name.
opacity	real	(Inherited from the <code>layer</code> class) The layer's master opacity (0.0 - 100.0).
visible	boolean	(Inherited from the <code>layer</code> class) Indication of whether the layer is visible.
background layer	boolean	Indication of whether the layer is a background layer.
fill opacity	real	The layer's interior opacity (0.0 - 100.0).
grouped	boolean	Indication of whether the layer is grouped with the layer below.
kind	Valid Values: brightness contrast layer channel mixer layer color balance layer curves layer gradient fill layer gradient map layer hue saturation layer inversion layer levels layer normal pattern fill layer posterize layer smart object layer selective color layer solid fill layer text layer threshold layer	Sets the layer's kind (such as 'text layer'). Note: You can use the <code>kind</code> property to make a background layer a normal layer; however, to make a layer a background layer, you must use background layer . Note: Valid only when the layer is empty and when <code>background layer</code> is false. See background layer .
pixels locked	boolean	Indicates whether the pixels in the layer's image can be edited using the paintbrush tool.
position locked	boolean	Indicates whether the pixels in the layer's image can be moved within the layer.
text object	text-object	Read-only. The text item that is associated with the art layer. Note: Valid only when <code>kind: text layer</code> .
transparent position locked	boolean	Indicates whether editing is confined to the opaque portions of the layer.

Valid Commands

[adjust](#)
[apply layer style](#)
[clear](#)

[copy](#)
[copy merged](#)
[count](#)
[cut](#)
[delete](#)
[duplicate](#)
[exists](#)
[filter](#)
[index](#)
[invert](#)
[link](#)
[make](#)
[merge](#)
[move](#)
[rasterize](#)
[rotate](#)
[scale](#)
[translate](#)
[unlink](#)

The following code creates a text layer by creating an art layer and setting its `kind` property to `text layer`, assigns the color to the variable `theTextColor`, and then uses the variable to set the text color to the RGB color.

```
tell application "Adobe Photoshop CS2"
    activate

    --create a variable named theDocRef
    --assign the current (active) document to it
    set theDocRef to the current document

    --create a variable for the text layer, create the layer as an art layer object
    --and use the kind property of the art layer object to make it a text layer
    set theTextLayer to make new art layer in theDocRef with
        properties {kind:text layer}

end tell
```

automatic contrast

The `automatic contrast` class supplies options properties for the [adjustment options](#) class to automatically adjust the overall contrast and mixture of colors in an RGB image.

The `automatic contrast` class, which is analogous to the Auto Contrast command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply an Auto Contrast adjustment to an image. The `automatic contrast` class can be used only in conjunction with the `adjust` command. It is not possible to get or create an `automatic contrast` object.

See the [adjust](#) command for additional details.

The following code sample demonstrates how to apply an Automatic Contrast adjustment using the `adjust` command of the `automatic contrast` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using automatic contrast adjustment-
with options {class:automatic contrast}
```

automatic levels

The `automatic levels` class supplies options properties for the [adjustment options](#) class to automatically adjust the black point and white point in an image.

The `automatic levels` class, which is analogous to the Auto Levels command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply an Auto Levels adjustment to an image. The `automatic levels` class can be used only in conjunction with the `adjust` command. It is not possible to get or create an `automatic levels` object.

See the [adjust](#) command for additional details.

The following code sample demonstrates how to apply an Automatic Levels adjustment using the `adjust` command of the `automatic levels` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using automatic levels adjustment-
with options {class:automatic levels}
```

batch options

Options to specify when running a Batch command.

Properties

Property	Value type	What it is
<code>destination</code>	Valid values: folder no destination save and close	The type of destination for the processed files (default: no destination).
<code>destination folder</code>	alias	The folder location for the processed files. Note: Valid only when <code>destination: folder</code> . See destination .
<code>error file</code>	alias	The file in which to log errors encountered. Note: To display errors on the screen (and stop batch processing when errors occur) leave blank.
<code>file naming</code>	Valid values: ddmm ddmmyy documentnamelower documentnamemixed documentnameupper extensionlower extensionupper mmdd mmddy serialletterlower serialletterupper serialnumber1 serialnumber2 serialnumber3 serialnumber4 yyddmm yymmdd yyyymmdd	A list of file naming options (maximum: 6). Note: Valid only when <code>destination: folder</code> . See destination .
<code>macintoshCompatible</code>	boolean	Indication of whether to make the final file nameS Macintosh compatible (default: true). Note: Valid only when <code>destination: folder</code> . See destination .
<code>overrideOpen</code>	boolean	Indication of whether to override action open commands (default: false).

Property	Value type	What it is (Continued)
<code>overrideSave</code>	boolean	Indication of whether to override save as action steps with the specified destination (default: <code>false</code>). Note: Valid only when <code>destination:folder</code> or <code>destination:save</code> and <code>close</code> . See destination .
<code>startingSerial</code>	number (long)	The starting serial number to use in naming files (default: 1). Note: Valid only when <code>destination:folder</code> . See destination .
<code>suppressOpen</code>	boolean	Indication of whether to suppress the file open options dialogs (default: <code>false</code>).
<code>suppressProfile</code>	boolean	Indication of whether to suppress the color profile warnings (default: <code>false</code>).
<code>typename</code>	string	Read-only. The class name of the referenced <code>batchOptions</code> object.
<code>unixCompatible</code>	boolean	Indication of whether to make the final file name Unix compatible (default: <code>true</code>). Note: Valid only when <code>destination:folder</code> . See destination .
<code>windowsCompatible</code>	boolean	Indication of whether to make the final file names Windows compatible (default: <code>true</code>). Note: Valid only when <code>destination:folder</code> . See destination .

Bitmap mode options

Options that can be specified when changing the document mode to Bitmap using the `change mode` command. The `bitmap mode` class inherits properties from the [change mode options](#) class.

This class is used to define a record containing properties used to specify options when changing a document's mode. `Bitmap mode options` can only be supplied in conjunction with the `change mode` command. It is not possible to get or create a `Bitmap mode options` object. See the [change mode](#) command for more details.

Properties

Properties	Value Type	What it is
<code>angle</code>	real	The angle (in degrees) at which to orient individual dots (-180 - 180). See screen shape . Note: Valid only when <code>conversion method: halftone screen</code> .
<code>conversion method</code>	Valid Values: custom pattern diffusion dither halftone screen conversion middle threshold pattern dither	The conversion method to use (default: diffusion dither).
<code>frequency</code>	real	The number of printer dots (per inch) to use (1.0 - 999.99). Note: Valid only when <code>conversion method: halftone screen</code> .
<code>pattern name</code>	Unicode text	The name of the pattern to use. Note: Valid only when <code>conversion method: custom pattern</code> .
<code>resolution</code>	real	The output resolution (in pixels per inch) (default: 72.0).
<code>screen shape</code>	Valid values: halftone cross halftone diamond halftone ellipse halftone line halftone round halftone square	The dot shape to use. Note: Valid only when <code>conversion method: halftone screen</code> .

blur

Options that may be supplied when applying a Blur filter.

The `blur` class supplies properties for the [filter options](#) class. The `blur` class is used only to define a record when using the `filter` command to apply a Blur filter. `blur` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `blur` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the blur filter using the `filter` command of the `blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using blur filter
with options {class:blur}
```

blur more

The `blur more` class supplies properties for the [filter options](#) class when using the `filter` command to apply a Blur More filter. The `blur more` class is used only to define a record when using the `filter` command to apply a Blur More filter. The `blur more` option can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `blur more` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the blur more filter using the `filter` command of the `blur more` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using blur more filter
with options {class:blur more}
```

BMP save options

Options that can be specified when saving a document in BMP format. See the [save](#) command for additional details. The `BMP save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a BMP file. `BMP save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `BMP save options` object.

Properties

Property	Value Type	What it is
<code>bits per sample</code>	Valid values: A1R5G5B5 A4R4G4B4 A8R8G8B8 eight four one R5G6B5 R8G8B8 sixteen thirty two twenty four X1R5G5B5 X4R4G4B4 X8R8G8B8	The number of bits per channel (default: twenty four).
<code>flipped row order</code>	boolean	Indication of whether to write the image from top to bottom (default: <code>false</code>). Note: Available only when <code>target operating system: windows</code> . See target operating system .
<code>RLE compression</code>	boolean	Indication of whether to use RLE compression. Note: Available only when <code>target operating system: windows</code> . See target operating system .
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.
<code>target operating system</code>	OS2 Windows	The target OS (default: Windows).

The following code creates the `myOptions` variable to hold the BMP save options, then uses the [save](#) command to save the document with the specified options. The code uses the default values for the properties `bits per channel` (twentyfour) and `target operating system` (true).

```
tell application "Adobe Photoshop CS2"
  make new document
  set myOptions to {class:BMP save options, -
    save layers: true, save alpha channels: true-
    image compression: RLE compression:true, save annotations: true}
  save current document in file myFile as BMP with options -
    myOptions appending no extension without copying
end tell
```

brightness and contrast

The `brightness and contrast` class supplies options properties for the [adjustment options](#) class to adjust the tonal range of an image.

The `brightness and contrast` class, which is analogous to the Brightness/Contrast command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when applying a Brightness/Contrast adjustment to an image. The `brightness and contrast` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `brightness and contrast` object.

See the [adjust](#) command for additional details.

Properties

Property	Value Type	What it is
<code>brightness level</code>	integer	(-100 - 100)
<code>contrast level</code>	integer	(-100 - 100)

The following code sample demonstrates how to apply a Brightness/Contrast adjustment using the `adjust` command of the `brightness and contrast` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using brightness and contrast
adjustment-
  with options {class:brightness and contrast}
```

camera raw open options

Options that can be specified when opening a document in Camera Raw format. The `camera raw open options` class inherits properties from the [open options](#) class.

This class is used to define a record containing properties that specify options when opening a document in raw format. `camera raw open options` can be supplied only in conjunction with the `open` command. It is not possible to get or create a `camera raw open options` object.

Properties

Property	Value type	What it is
<code>bitsPerChannel</code>	Valid values: EIGHT ONE SIXTEEN THIRTYTWO	The number of bits per channel.
<code>blueHue</code>	integer	The blue hue of the shot (-100 - 100).
<code>blueSaturation</code>	integer	The blue saturation of the shot (-100 - 100).
<code>brightness</code>	integer	The brightness of the shot (0 - 150).
<code>chromaticAberrationBY</code>	integer	The chromatic aberration B/Y of the shot (-100 - 100).
<code>chromaticAberrationRC</code>	integer	The chromatic aberration R/C of the shot (-100 - 100).
<code>colorNoiseReduction</code>	integer	The color noise reduction of the shot (0 - 100).
<code>colorSpace</code>	Valid values: ADOBERGB COLORMATCHRGB PROPHOTORGB SRGB	The colorspace for the image.
<code>contrast</code>	integer	The contrast of the shot (-50 - 100).
<code>exposure</code>	real	The exposure of the shot (4.0 - 4.0).
<code>greenHue</code>	integer	The green hue of the shot (-100 - 100).
<code>greenSaturation</code>	integer	The green saturation of the shot (-100 - 100).
<code>luminanceSmoothing</code>	integer	The luminance smoothing of the shot (0 - 100).
<code>redHue</code>	integer	The red hue of the shot (-100 - 100).
<code>redSaturation</code>	integer	The red saturation of the shot (-100 - 100).
<code>resolution</code>	real	The resolution of the document in pixels per inch (1 - 999).
<code>saturation</code>	integer	The saturation of the shot (-100 - 100).

Property	Value type	What it is (Continued)
settings	Valid values: CAMERA CUSTOM SELECTEDIMAGE	The global settings for all Camera RAW options.
shadows	integer	The shadows of the shot (0 - 100).
shadowTint	integer	The shadow tint of the shot (-100 - 100).
sharpness	integer	The sharpness of the shot (0 - 100).
size	Valid values: EXTRALARGE LARGE MAXIMUM MEDIUM MINIMUM SMALL	The size of the new document.
temperature	integer	The temperature of the shot (2000 - 50000).
tint	integer	The tint of the shot (-150 - 150).
typename	string	Read-only. The class name of the referenced <code>cameraRawOpenOptions</code> object.
vignettingAmount	integer	The vignetting amount of the shot (-100 - 100).
vignettingMidpoint	integer	The vignetting mid point of the shot (-100 - 100).
whiteBalance	Valid values: ASSHOT AUTO CLOUDY CUSTOM DAYLIGHT FLASH FLUORESCENT SHADE TUNGSTEN	The white balance options for the image.

The following code opens a camera raw document.

```
tell application "Adobe Photoshop CS2"
    set myFilePath to alias "Applications:CRAWFiles:MyFile.jpeg"
    open myFilePath as camera raw with options -
        {class:camera raw open options, bits per channel:eight, -
        color noise reduction:20, temperature:47, -
        white balance:cloudy}
end tell
```

change mode options

Options that can be supplied when changing a document's mode using the `change mode` command. See [change mode](#) for additional details.

The following classes inherit properties from the `change mode options` class: [Bitmap mode options](#), [indexed mode options](#).

Valid Commands

[change mode](#)

channel, channels

A channel in a document. Can be either a component channel representing a color of the document color model or an alpha channel

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	All of this objects properties returned in a single record.
channel color	color value	The channel's color. Note: Not valid when <code>kind: component channel</code> . See kind .
container	reference	The object's container.
histogram	list of integers	Read-only. A histogram of color values for the channel. Note: Not valid when <code>kind: component channel</code> . For component channel histogram values, use the <code>histogram</code> property of the document object instead. See 'histogram' on page 48 .
kind	Valid values: component channel masked area channel selected area channel spot color channel	The type of channel.
name	Unicode text	The channel's name.
opacity	real	The opacity (as percent) to use for alpha channels or the solidity to use for spot channels (0 - 100). Note: Valid only when <code>kind: masked area</code> or <code>kind: selected area</code> .
visible	boolean	Indication of whether the channel is visible.

Valid Commands

[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[index](#)

[make](#)
[merge](#)

clouds

Options that may be supplied when applying a Clouds filter.

The `clouds` class supplies properties for the [filter options](#) class. The `clouds` class is used only to define a record when using the `filter` command to apply a Clouds filter. `clouds` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `clouds` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the clouds filter using the `filter` command of the `clouds` class (inherited from the `filter options` super class).

```
...  
filter current layer of the current document using clouds filter  
    with options {class:clouds}
```

CMYK color

Options that can be specified for a CMYK color: numeric values for the component colors. The `CMYK color` class inherits properties from the `color value` class.

This class a property of the `color value` object and is used to define a record that specifies options when creating a color value.

Note: If the `mode` of a document is RGB and you specify the color value for a stroke, foreground, or background color property using a `CMYK color` object, Adobe Photoshop CS2 translates the CMYK color specification into an RGB color specification. The same thing happens if the document's color mode is CMYK and you specify colors using a `RGB color` object. Since this translation can cause information loss you should specify colors using the color class that matches the document's color mode.

Properties

Property	Value Type	What it is
<code>cyan</code>	real	The cyan color value (as percent) (0.0 - 100.0).
<code>magenta</code>	real	The magenta color value (as percent) (0.0 100.0).
<code>yellow</code>	real	The yellow color value (as percent) (0.0 - 100.0).
<code>black</code>	real	The black color value (as percent) (0.0 - 100.0).

The following code creates a color value using the `RGB color` class, assigns the color to the variable `theTextColor`, and then uses the variable to set the text color to the RGB color.

```
tell application "Adobe Photoshop CS2"
  activate

  --create a variable named theDocRef
  --assign the current (active) document to it
  set theDocRef to the current document

  --create a variable that contains a color object of the RGB color class
  --whose color is red
  set theTextColor to {class:RGB color, red:255, green:0, blue:0}

  --create a variable for the text layer, create the layer as an art layer object
  --and use the kind property of the art layer object to make it a text layer
  set theTextLayer to make new art layer in theDocRef with-
    properties {kind:text layer}

  --Set the contents, size, position and color of the text layer
  set contents of text object of theTextLayer to "Hello, World!"
  set size of text object of theTextLayer to 36
  set position of text object of theTextLayer to {0.75, 1}
  set stroke color of text object of theTextLayer to theTextColor
end tell
```

color balance

The `color balance` class specifies options for the [adjustment options](#) class to change the overall mixture of colors in an image for generalized color correction.

The `color balance` class, which is analogous to the Color Balance command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when applying a Color Balance adjustment to an image. The `color balance` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `color balance` object.

See the [adjust](#) command for additional details.

Properties

Property	Value Type	What it is
<code>shadows</code>	list of integers	The list of adjustments for shadows (three values (-100 - 100 required). The specified values indicate the color levels for component colors of the document mode, either RGB or CMYK. For CMYK mode, the values represent C, M, and Y.
<code>midtone</code> s	list of integers	The list of adjustments for midtones (three values (-100 - 100 required). The specified values indicate the color levels for component colors of the document mode, either RGB or CMYK. For CMYK mode, the values represent C, M, and Y.
<code>highlights</code>	list of integers	The list of adjustments for highlights (three values (-100 - 100 required). The specified values indicate the color levels for component colors of the document mode, either RGB or CMYK. For CMYK mode, the values represent C, M, and Y.
<code>preserve luminosity</code>	boolean	Indication of whether to preserve the image luminosity.

The following code sample demonstrates how to apply a color balance adjustment using the `adjust` command of the `color balance` class (inherited from the `adjustment options` super class). This code adjusts only the shadows in the image.

```
...
adjust current layer of the current document using color balance adjustment-
  with options {class:color balance, shadows: {45, 4,5}, preserve luminosity:false}
```

color value

A color value. The following classes inherit properties from the `color value` class: [CMYK color](#), [gray color](#), [HSB color](#), [Lab color](#), [no color](#), [RGB color](#), [RGB hex color](#).

The following script creates an RGB color value to use as the fill color for the selection.

```
tell application "Adobe Photoshop CS2"
  make new art layer at beginning of current document -
    with properties {name:"MyBlendLayer", blend mode:normal}
  select all current document
  fill selection of current document with contents -
    {class:RGB color, red:255, green:0, blue:0}
end tell
```

Valid Commands

[convert color](#)
[equal colors](#)
[web safe color](#)

contact sheet options

Options that can be specified for a contact sheet.

Properties

Property	Value Type	What it is
across first	boolean	Indication of whether to place the images horizontally (left to right, then top to bottom) first (default: <code>true</code>).
best fit	boolean	Indication of whether to rotate images for the best fit (default: <code>false</code>).
caption	boolean	Indication of whether to use the filename as a caption for the image (default: <code>true</code>).
column count	number (long)	The number of columns to include (1 - 100; default: 5).
flatten	boolean	Indication of whether to flatten all layers in the final document (default: <code>true</code>).
font	Valid values: arial couriernew helvetica timesnewroman	The font used for the caption (default: <code>arial</code>).
font size	number (long)	The font size to use for the caption (default: 12).
height	number (long)	The height (in pixels) of the resulting document (100 - 2900; default: 720).
horizontal	number (long)	The horizontal spacing (in pixels) between images (0 - 29000; default: 1).
mode	Valid values: bitmap CMYK grayscale lab RGB	The document color mode (default: <code>RGB</code>).
resolution	number (double)	The resolution of the document in pixels per inch (35 - 1200; default: 72.0).
row count	number (long)	The number of rows to use (1 - 100; default: 6).
typename	string	Read-only. The class name of the referenced object.
use auto spacing	boolean	Indication of whether to auto space the images (default: <code>true</code>).

Property	Value Type	What it is (Continued)
vertical	number (long)	The vertical spacing (in pixels) between images (0 - 29000; default: 1). Note: Valid only when use auto spacing: false.
width	number (long)	The width (in pixels) of the resulting document (100 - 2900; default: 576).

curves

The `curves` class supplies options for the [adjustment options](#) class to adjust the tonal range of an image. You can also use the `curves` object to make precise adjustments to individual color channels within an image.

The `curves` class, which is analogous to the Curves command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when applying a Curves adjustment to an image. The `curves` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `curves` object.

See the [adjust](#) command for additional details.

Properties

Property	Value Type	What it is
<code>curve points</code>	list of lists	The list of lists of coordinates (x and y) that indicate the points that describe the curve. (You must include 2 - 14 points). Tip: For more information, please refer to Adobe Photoshop CS2 Help.

The following code sample demonstrates how to apply a Curves adjustment using the `adjust` command of the `curves` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using curves adjustment-
  with options {class:curves, curve points:{45, 4}, {42, 9}, {40, 17}{36, 22}}
```

custom filter

Options that may be supplied when applying a Custom filter.

The `custom filter` class supplies properties for the [filter options](#) class. The `custom filter` class is used only to define a record whose properties specify options when applying a Custom filter. `custom filter options` can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `custom filter` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
characteristic	list of integers	The filter characteristics (array of 25 values; corresponds to a left to right, top to bottom traversal of the array presented in the user interface). Note: Required parameter values define the filter. Refer to Adobe Photoshop CS2 Help for specific instructions.
scaling	integer	
offset	integer	

The following code sample demonstrates how to apply a custom filter using the `filter` command of the `custom filter` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using custom filter ~
  with options {class:add noise, distribution:uniform ~
    monochromatic:false}
```

deinterlace

Options that may be supplied when applying a De-Interlace filter.

The `deinterlace` class supplies properties for the [filter options](#) class. The `deinterlace` class is used only to define a record whose properties specify options when applying a De-Interlace filter. `deinterlace` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `deinterlace` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>eliminate</code>	Valid values: even fields odd fields	The type of fields to eliminate.
<code>create new fields by</code>	Valid values: duplication interpolation	The method to use to create new fields.

The following code sample demonstrates how to apply the De-Interlace filter using the `filter` command of the `deinterlace` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using deinterlace filter -
  with options {class:deinterlace, eliminate:odd fields-
    create new fields by:duplication}
```

desaturate

The `desaturate` class supplies options for the [adjustment options](#) class to convert a color image to a grayscale image in the current color mode by assigning equal values of each component color to each pixel.

The `desaturate` class, which is analogous to the Desaturate command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply a desaturation adjustment to a selected layer. The `desaturate` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `desaturate` object.

See the [adjust](#) command for additional details.

The following code sample demonstrates how to apply a Desaturate adjustment using the `adjust` command of the `desaturate` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using desaturate adjustment-
with options {class:desaturate}
```

despeckle

Options that may be supplied when applying a Despeckle filter.

The `despeckle` class supplies properties for the [filter options](#) class. The `despeckle` class is used only to define a record whose properties specify options when using the `filter` command to apply a Despeckle filter. `despeckle` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `despeckle` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the despeckle filter using the `filter` command of the `despeckle` filter class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using despeckle filter
    with options {class:despeckle}
```

difference clouds

Options that may be supplied when applying a Difference Clouds filter.

The `difference clouds` class supplies properties for the [filter options](#) class. The `difference clouds` class is used only to define a record whose properties specify options when using the `filter` command to apply a Difference Clouds filter. `difference clouds` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `difference clouds` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the difference clouds filter using the `filter` command of the `difference clouds` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using difference clouds filter
    with options {class:difference clouds}
```

diffuse glow

Options that may be supplied when applying a Diffuse Glow filter.

The `diffuse glow` class supplies properties for the [filter options](#) class. The `diffuse glow` class is used only to define a record whose properties specify options when applying a Diffuse Glow filter. `diffuse glow` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `diffuse glow` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>graininess</code>	integer	(0 - 10)
<code>glow amount</code>	integer	(0 - 20)
<code>clear amount</code>	integer	(0 - 20)

The following code sample demonstrates how to apply the diffuse glow filter using the `filter` command of the `diffuse glow` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using clouds filter
    with options {class:clouds}
```

displace filter

Options that may be supplied when applying a Displace filter.

The `displace filter` class supplies properties for the [filter options](#) class. The `displace filter` class is used only to define a record whose properties specify options when applying a Displace filter. `displace filter` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `displace filter` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>horizontal scale</code>	integer	(-999 - 999)
<code>vertical scale</code>	integer	(-999 - 999)
<code>kind</code>	Valid values: stretch to fit tile	The displacement map type.
<code>undefined areas</code>	Valid values: repeat edge pixels wrap around	The method to use to treat undistorted areas.
<code>displacement map definition</code>	alias	The location of the desired displacement map.

The following code sample demonstrates how to apply the Displace filter using the `filter` command of the `displace` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using displace filter ↵
  with options {class:displace, horizontal scale:250↵
    vertical scale:350, kind:tile, undefined areas:repeat edge pixels↵
    displacement map definition:"Hard Disk:Photoshop CS 2:Displacement Maps:DM1"}
```

document, documents

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.

Elements

Element	Refer to by
art layer	name, numeric index, as a range of elements, before after another element, satisfying a test
channel	name, numeric index, as a range of elements, before after another element, satisfying a test
history state	name, numeric index, as a range of elements, before after another element, satisfying a test
layer comp	name, numeric index, as a range of elements, before after another element, satisfying a test
layer	name, numeric index, as a range of elements, before after another element, satisfying a test
layer set	name, numeric index, as a range of elements, before after another element, satisfying a test
path item	name, numeric index, as a range of elements, before after another element, satisfying a test

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	The object's properties returned in a single record.
background layer	art layer	Read-only. The background layer for the document. Note: Valid only for documents that have a background layer.

Property	Value Type	What it is (Continued)
<code>bits per channel</code>	Valid values: eight one sixteen	The number of bits per channel.
<code>color profile kind</code>	Valid values: custom none working	The type of color profile management for the document.
<code>color profile name</code>	Unicode text	The name of color profile for the document. Note: Valid only when <code>color profile kind</code> : <code>custom</code> OR <code>color profile kind</code> : <code>working</code> .
<code>component channels</code>	list of channels	Read-only. A list of the component color channels.
<code>current channels</code>	list of channel	The selected channels.
<code>current history brush source</code>	history state	The current history state to use with the history brush.
<code>current history state</code>	history state	The selected history state.
<code>current layer</code>	layer	The selected layer.
<code>file path</code>	alias	Read-only. The full path name of the document.
<code>height</code>	real	Read-only. The document height (unit value).
<code>histogram</code>	list of 256 integers	Read-only. A histogram of values for the composite document. Note: Valid only when <code>mode</code> : <code>RGB</code> ; <code>mode</code> : <code>CMYK</code> ; OR <code>mode</code> : <code>indexed color</code> . See mode .
<code>info</code>	info-object	Read-only. The document metadata.
<code>initial fill</code>	Valid values: transparent use background color white	Read-only. The initial fill. Note: Valid only when used as an option with the 'make new document' command. See make .
<code>managed</code>	boolean	Read-only. Indication of whether the document is a workgroup document.

Property	Value Type	What it is (Continued)
mode	Valid values: bitmap CMYK duotone grayscale indexed color Lab multichannel RGB	Read-only. The color profile for the document.
modified	boolean	Read-only. Indication of whether the document has been modified since the last save.
name	Unicode text	Read-only. The document's name.
pixel aspect ratio	real	The (custom) pixel aspect ratio to use (0.100 - 10.000).
quick mask mode	boolean	Indication of whether the document is in the quick mask mode.
resolution	real	Read-only. The document's resolution (in pixels per inch).
selection	selection-object	Read-only. The selected area of the document.
width	real	Read-only. The document width (unit value).
XMP metadata	XMP metadata	Read-only. Camera raw settings for the image. Note: Valid only for documents opened in Camera Raw format.

Valid Commands

[change mode](#)
[close](#)
[convert](#)
[count](#)
[crop](#)
[delete](#)
[deselect](#)
[duplicate](#)
[exists](#)
[expand](#)
[export](#)
[flatten](#)
[flip canvas](#)
[import annotations](#)
[index](#)
[make](#)
[merge visible layers](#)
[paste](#)
[print](#)
[resize canvas](#)
[resize image](#)
[reveal all](#)
[rotate canvas](#)

[save](#)
[split channels](#)
[trap](#)
[trim](#)

dust and scratches

Options that may be supplied when applying a Dust & Scratches filter.

The `dust and scratches` class supplies properties for the [filter options](#) class. The `dust and scratches` class is used only to define a record whose properties specify options when applying a Dust & Scratches filter. `dust and scratches` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `dust and scratches` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>radius</code>	integer	(1 - 100) in pixels.
<code>threshold</code>	integer	(0 - 255)

The following code sample demonstrates how to apply the Dust & Scratches filter using the `filter` command of the `dust and scratches` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using dust and scratches filter -
  with options {class:dust and scratches, radius:15-
    threshold:212}
```

EPS open options

Settings related to opening a generic EPS document. The `EPS open options` class inherits properties from the [open options](#) class.

This class is used to define a record containing properties that specify options when opening a document in EPS format. `EPS open options` can be supplied only in conjunction with the `open` command. It is not possible to get or create a `EPS open options` object.

Properties

Property	Value Type	What it is
constrain proportions	boolean	Indication of whether to constrain the image proportions.
height	real	The image height (unit value).
mode	Valid values: CMYK grayscale Lab RGB	The color profile to use as the document mode.
resolution	real	The resolution of the document (in pixels per inch).
use antialias	boolean	Indication of whether to use antialias.
width	real	The image width (unit value).

The following code opens an EPS document.

```
tell application "Adobe Photoshop CS2"
  set myFilePath to alias "Applications:EPSFiles:MyFile.eps"
  open myFilePath as EPS with options -
    {class:EPS open options, height:pixels 100, -
      width:pixels 200, mode:RGB, resolution:72, -
      use antialias:true, constrain proportions:true}
end tell
```

EPS save options

Options that can be specified when saving a document in EPS format. See the [save](#) command for additional details. The `EPS save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as an EPS file. `EPS save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create an `EPS save options` object.

Properties

Property	Value Type	What it is
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>encoding</code>	Valid Values: ASCII binary high quality JPEG low quality JPEG maximum quality JPEG medium quality JPEG	The type of encoding to use (default: <code>binary</code>).
<code>halftone screen</code>	boolean	Indication of whether to include the halftone screen (default: <code>false</code>).
<code>image interpolation</code>	boolean	Indication of whether to use image interpolation (default: <code>false</code>).
<code>PostScript color management</code>	boolean	Indication of whether to use Postscript color management (default: <code>false</code>).
<code>preview type</code>	Valid Values: eight bit Mac OS eight bit TIFF JPEG Mac OS monochrome Mac OS monochrome TIFF none	The preview type (default: <code>monochrome TIFF</code>).
<code>transfer function</code>	boolean	Indication of whether to include the Transfer functions to compensate for dot gain between the image and film (default: <code>false</code>).
<code>transparent whites</code>	boolean	Indication of whether to display white areas as transparent. Note: Valid only when saving BitMap documents.
<code>vector data</code>	boolean	Indication of whether to include vector data. Note: Valid only if the document includes vector data (text).

The following code creates the `myOptions` variable to hold the `EPS save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document
  set myOptions to {class:EPS save options, -
    embed color profile: true}
  save current document in file myFile as Photoshop EPS with options -
    myOptions appending no extension without copying
end tell
```

equalize

The `equalize` class supplies options for the [adjustment options](#) class to convert a color image to redistribute the brightness values of the pixels in an image so that they more evenly represent the entire range of brightness levels.

The `equalize` class, which is analogous to the Equalize command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply an Equalize adjustment to a selected area. The `equalize` class can be used only in conjunction with the `adjust` command. It is not possible to get or create an `equalize` object.

See the [adjust](#) command for additional details.

The following code sample demonstrates how to apply a Equalize adjustment using the `adjust` command of the `equalize` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using equalize adjustment-
  with options {class:equalize, curve points:{45, 4}, {42, 9}, {40, 17}{36, 22}}
```

export options save for web

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

Properties

Property	Value type	What it is
<code>blur</code>	real	Applies blur to the image to reduce artifacts (default: 0.0).
<code>colorReduction</code>	Valid values: adaptive blackwhite custom grayscale macintosh perceptual restrictive selective windows	The color reduction algorithm (default: selective).
<code>colors</code>	integer	The number of colors in the palette (default: 256).
<code>dither</code>	Valid values: diffusion noise none pattern	The type of dither (default: diffusion).
<code>ditherAmount</code>	integer	The amount of dither (default: 100). Note: Valid only when <code>dither:diffusion</code> . See dither .
<code>format</code>	Valid values: aliaspix bmp compuservegif electricimage jpeg pcx photoshop photoshopdcs_1 photoshopdcs_2 photoshopeps photoshoppdf pictfileformat pictresourceformat pixar png portablebitmap raw scitext sgirgb softimage targa tiff wavefrontrla wirelessbitmap	The file format to use (default: compuservegif).

Property	Value type	What it is (Continued)
<code>includeProfile</code>	boolean	Indication of whether to include the document's embedded color profile (default: <code>false</code>).
<code>interlaced</code>	boolean	Indication of whether to download in multiple passes; progressive (default: <code>false</code>).
<code>lossy</code>	integer	The amount of lossiness allowed (default: 0).
<code>matteColor</code>	RGB color	The colors to blend transparent pixels against.
<code>optimized</code>	boolean	Indication of whether to create smaller but less compatible files (default: <code>true</code>). Note: Valid only when <code>format</code> : JPEG. See format .
<code>PNG8</code>	boolean	Indicates the number of bits; <code>true</code> = 8, <code>false</code> = 24 (default: <code>true</code>). Note: Valid only when <code>format</code> : PNG. See format .
<code>quality</code>	integer	The quality of the produced image (0 - 100 as percentage; default: 60).
<code>transparency</code>	boolean	Indication of transparent areas of the image should be included in the saved image (default: <code>true</code>).
<code>transparencyAmount</code>	integer	The amount of transparency dither (default: 100). Note: Valid only if <code>transparency</code> : <code>true</code> . See transparency .
<code>transparencyDither</code>	Valid values: diffusion noise none pattern	The transparency dither algorithm (default: none).
<code>typename</code>	string	Read-only. The class name of the referenced <code>ExportOptionsSaveForWeb</code> object.
<code>webSnap</code>	integer	The tolerance amount within which to snap close colors to web palette colors (default: 0).

filter options

Options used with the `filter` command.

The following classes inherit properties from the [filter options](#) class: [add noise](#), [blur](#), [blur more](#), [clouds](#), [custom filter](#), [deinterlace](#), [despeckle](#), [difference clouds](#), [diffuse glow](#), [displace filter](#), [dust and scratches](#), [Gaussian blur](#), [glass filter](#), [high pass](#), [high pass](#), [lens blur](#), [maximum filter](#), [median noise](#), [minimum filter](#), [motion blur](#), [notifier](#), [notifiers](#), [ocean ripple](#), [offset filter](#), [pinch](#), [polar coordinates](#), [radial blur](#), [ripple](#), [sharpen](#), [sharpen edges](#), [sharpen more](#), [shear](#), [smart blur](#), [spherize](#), [texture fill](#), [twirl](#), [unsharp mask](#), [wave filter](#), [zigzag](#).

Valid Commands

[filter](#)

font, fonts

An installed font. The `font` object is an element of the `application` class. You use the `font` object to retrieve information about the fonts installed on your computer. See [‘application’ on page 13](#) for more information.

Properties

Property	Value Type	What it is
<code>best type</code>	type class	Read-only. The best type for the object's value.
<code>class type</code>	class	The object's class.
<code>default type</code>	type class	Read-only. The default type for the object's value.
<code>index</code>	integer	Read-only. The index of this instance of the object.
<code>properties</code>	record	All of this objects properties returned in a single record.
<code>family</code>	Unicode text	Read-only. The font family.
<code>name</code>	Unicode text	The text face name.
<code>PostScript name</code>	Unicode text	The font's PostScript name.
<code>style</code>	Unicode text	The font's style name.

The following code uses the standard AppleScript commands `get` and `prompt` to display a dialog that lists the fonts installed on the computer running the script.

```
tell application "Adobe Photoshop CS2"
    set fontsInstalled to get fonts
    prompt (fontsInstalled)
end tell
```

Valid commands:

[count](#)
[exists](#)
[index](#)

gallery banner options

Options that can be specified for a Web photo gallery banner.

This class a property of the [gallery options](#) object and is used to define a record that specifies options when creating a Web photo gallery. Gallery banner options can only be supplied in conjunction with the `create photo gallery` command. It is not possible to get or create a Gallery banner options object.

See the [create photo gallery](#) command for additional details.

Properties

Property	Value Type	What it is
<code>contact info</code>	Unicode text	Web photo gallery contact info.
<code>date</code>	Unicode text	Web photo gallery date (default: current date).
<code>font</code>	Valid values: Arial Courier New Helvetica Times New Roman	The font setting for the banner text (default: Arial).
<code>font size</code>	integer	The font size for the banner text (1 - 7; default: 3).
<code>photographer</code>	Unicode text	Web photo gallery photographer.
<code>site name</code>	Unicode text	Web photo gallery site name (default: Adobe Web Photo Gallery).

gallery color options

Color options that can be specified for a Web photo gallery.

This class is a property of the [gallery options](#) object and is used to define a record that specifies options when creating a Web photo gallery. Gallery color options can only be supplied in conjunction with the `create photo gallery` command. It is not possible to get or create a `Gallery color options` object.

See the [create photo gallery](#) command for additional details.

Properties

Property	Value Type	What it is
<code>active link color</code>	RGB color	The color used to indicate an active link.
<code>background color</code>	RGB color	The background color.
<code>banner color</code>	RGB color	The banner color.
<code>link color</code>	RGB color	The color used to indicate a link.
<code>text color</code>	RGB color	The text color.
<code>visited link color</code>	RGB color	The color used to indicate a visited link.

gallery images options

Options for the larger images (not thumbnails) in a Web photo gallery.

This class a property of the [gallery options](#) object and is used to define a record that specifies options when creating a Web photo gallery. Gallery images options can only be supplied in conjunction with the `create photo gallery` command. It is not possible to get or create a Gallery images options object.

See the [create photo gallery](#) command for additional details.

Note: See [gallery thumbnail options](#) for information on specifying thumbnail image options.

Properties

Property	Value Type	What it is
<code>border</code>	integer	The width (in pixels) of the border between images (0 - 99; default: 0).
<code>caption</code>	boolean	Indication of whether to generate image captions (default: <code>false</code>).
<code>dimension</code>	integer	The dimensions (in pixels) of a resized image (default: 350).
<code>font</code>	Valid values: Arial Courier New Helvetica Times New Roman	The font to use for captions (default: Arial).
<code>font size</code>	integer	The font size to use for captions (1 - 7; default: 3).
<code>image quality</code>	integer	The quality setting for JPEG images (0 - 12; default: 5).
<code>include copyright</code>	boolean	Indication of whether to include copyright information in captions (default: <code>false</code>).
<code>include credits</code>	boolean	Indication of whether to include credits information in captions (default: <code>false</code>).
<code>include file name</code>	boolean	Indication of whether to include the file name in captions (default: <code>false</code>).
<code>include title</code>	boolean	Indication of whether to include the title in captions (default: <code>false</code>).
<code>numeric links</code>	boolean	Indication of whether to add numeric links (default: <code>true</code>).
<code>resize constraint</code>	Valid values: constrain both constrain height constrain width	Indication of whether and how the image should be constrained (default: <code>constrain both</code>).
<code>resize images</code>	boolean	Indication of whether to automatically resize images for placement on the gallery pages (default: <code>true</code>).

gallery options

Options for the `create photo gallery` command.

Properties

Property	Value Type	What it is
<code>banner options</code>	gallery banner options	Options related to banner settings.
<code>custom color options</code>	gallery color options	Options related to custom color settings.
<code>email address</code>	Unicode text	The email address to show on the web page.
<code>images options</code>	gallery images options	Options related to images settings.
<code>layout style</code>	Unicode text	The style to use for laying out the web page (default: <code>Centered Frame 1 - Basic</code>).
<code>preserve all metadata</code>	boolean	Indication of whether to save the metadata in JPEG files (default: <code>false</code>).
<code>security options</code>	gallery security options	Options related to security settings.
<code>short extension</code>	boolean	Indication of whether to use the short (.htm), as opposed to the long (.html) web page file name extension (default: <code>true</code>).
<code>size attributes</code>	boolean	Indication of whether to add width and height attributes for images (default: <code>true</code>).
<code>sub folders</code>	boolean	Indication of whether to include files found in the in sub folders of the input folder (default: <code>true</code>).
<code>thumbnail options</code>	gallery thumbnail options	Options related to thumbnail settings.
<code>UTF8 encoding</code>	boolean	Indication of whether the web page should use UTF-8 encoding (default: <code>false</code>).

The following script creates a Web photo gallery.

```
tell application "Adobe Photoshop CS2"
  create photo gallery with options: {from file:"HardDisk:Adobe Photoshop
CS2:WbGalleries", -
  to file:"HardDisk:Adobe Photoshop CS2:NewWbGalleries", -
  email address:"photoGallery@mygallery.com",-
  class:gallery banner options {photographer:"Tom O'Rourke", -
  contact info:"torourke@torourke.com", -
  font:times new roman, sitename:"www.torourke.com"}-
  class: gallery color options: {active link color:blue,-
  background color:white, banner color: teal,
  link color: RED, text color:black, visited link color:purple}-
  class: gallery security options { content: gallery custom text, custom text:
copyright 2004 TOR-
```

```
        font:courier new, text position: gallery centered; text rotate: counter  
clockwise 45}}  
    end tell
```

Valid Commands

[create photo gallery](#)

gallery security options

Options for security text in a Web photo gallery.

This class a property of the [gallery options](#) object and is used to define a record that specifies options when creating a Web photo gallery. Gallery security options can only be supplied in conjunction with the `create photo gallery` command. It is not possible to get or create an `Gallery security options` object.

See the [create photo gallery](#) command for additional details.

Properties

Property	Value Type	What it is
<code>content</code>	Valid values: gallery caption gallery copyright gallery credit gallery custom text gallery filename gallery none gallery title	The content to use for Web photo gallery security text (default: <code>gallery none</code>).
<code>custom text</code>	Unicode text	Custom text to use for Web photo gallery security text. Note: Valid only when <code>content: custom text</code> . See content .
<code>font</code>	Arial Courier New Helvetica Times New Roman	The font to use for Web photo gallery security text (default: <code>Arial</code>).
<code>font size</code>	integer	The font size to use for Web photo gallery security text (minimum 1; default: 36).
<code>opacity</code>	integer	The opacity (as percent) to use for web page security text (default: 100).
<code>text color</code>	RGB color	The web page security text color.
<code>text position</code>	Valid values: gallery centered gallery lower left gallery lower right gallery upper left gallery upper right	The position of web page security text (default: <code>gallery centered</code>).
<code>text rotate</code>	Valid values: clock wise 45 clock wise 90 counter clock wise 45 counter clock wise 90 zero	The orientation to use for web page security text (default: <code>zero</code>).

gallery thumbnail options

Options for the thumbnail images in a Web photo gallery.

This class a property of the [gallery options](#) object and is used to define a record that specifies options when creating a Web photo gallery. `Gallery thumbnail options` can only be supplied in conjunction with the `create photo gallery` command. It is not possible to get or create an `Gallery thumbnail options` object.

See the [create photo gallery](#) command for additional details.

Properties

Property	Value Type	What it is
<code>border</code>	integer	The width (in pixels) of the border around thumbnail images (0 - 99; default: 0).
<code>caption</code>	boolean	Indication of whether to display captions (default: <code>false</code>).
<code>column count</code>	integer	The number of columns on the page (default: 5).
<code>dimension</code>	integer	Thumbnail dimensions in pixels (default: 75).
<code>font</code>	Valid values: Arial Courier New Helvetica Times New Roman	The font to use (default: <code>Arial</code>).
<code>font size</code>	integer	The font size for thumbnail images text (1 - 7; default: 3).
<code>include copyright</code>	boolean	Indication of whether to include copyright information for thumbnails (default: <code>false</code>).
<code>include credits</code>	boolean	Indication of whether to include credits for thumbnails (default: <code>false</code>).
<code>include file name</code>	boolean	Indication of whether to include file names for thumbnails (default: <code>false</code>).
<code>include title</code>	boolean	Indication of whether to include titles for thumbnails (default: <code>false</code>).
<code>row count integer</code>	integer	The number of rows on the page (default: 3).
<code>size</code>	Valid values: gallery custom gallery large gallery medium gallery small	The thumbnail image size (default: <code>gallery medium</code>).

Gaussian blur

Options that may be supplied when applying a Gaussian Blur filter.

The `Gaussian blur` class supplies properties for the [filter options](#) class. The `Gaussian blur` class is used only to define a record whose properties specify options when applying a Gaussian Blur filter.

`Gaussian blur` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `Gaussian blur` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>radius</code>	real	The radius in pixels in which to apply the filter.

The following code sample demonstrates how to apply the Gaussian Blur filter using the `filter` command of the `Gaussian blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using Gaussian blur filter -
    with options {class:Gaussian blur, radius:200}
```

GIF save options

Options that can be specified when saving a document in GIF format. See the [save](#) command for additional details. The `GIF save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a GIF file. `GIF save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `GIF save options` object.

Properties

Property	Value Type	What it is
<code>colors in palette</code>	integer	The number of colors in the Color palette. Note: Valid only when <code>palette: local adaptive</code> ; <code>palette: local perceptual</code> ; <code>palette: local selective</code> ; <code>palette: Mac OS system</code> ; <code>palette: uniform</code> ; <code>palette: web</code> ; or <code>palette: Windows system</code> . See palette .
<code>dither</code>	Valid values: diffusion noise none pattern	The dither type to use.
<code>dither amount</code>	integer	The amount of dither to use (1 - 100). Note: Valid only when <code>dither: diffusion</code> . See dither .
<code>forced colors</code>	Valid values: black and white none primaries web	Forces the inclusion of specified colors.
<code>interlaced</code>	boolean	Indication of whether to interlace rows (default: <code>false</code>).
<code>matte background color</code>	Valid values: matte black matte foreground color matte Netscape gray none semi gray white matte	The color to use to fill antialiased edges adjacent to transparent areas of the image (default: <code>white matte</code>). Note: When <code>transparency: false</code> , the matte color is applied to transparent areas. See transparency .

Property	Value Type	What it is (Continued)
palette	Valid values: exact local adaptive local perceptual local selective Mac OS system master adaptive master perceptual master selective previous uniform web Windows system	The type of palette to use (default: <code>exact</code>).
preserve exact colors	boolean	Indication of whether to protect colors in the image that contain entries in the color table from being dithered. Note: Valid only when <code>dither: diffusion</code> . See dither .
transparency	boolean	Indication of whether to preserve transparent areas of the image during conversion to GIF format.

The following code creates the `myOptions` variable to hold the GIF save options, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document
  set myOptions to {class:GIF save options, -
    dither: diffusion, dither amount: 46-
    preserve exact colors: true, matte background color: foreground color matte}
  save current document in file myFile as Compuserve GIF with options -
    myOptions appending no extension without copying
end tell
```

glass filter

Options that may be supplied when applying a Glass filter.

The `glass filter` class supplies properties for the [filter options](#) class. The `glass filter` class is used only to define a record whose properties specify options when applying a Glass filter. `glass filter` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `glass filter` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>distortion</code>	integer	(0 - 20)
<code>smoothness</code>	integer	(1 - 15)
<code>scaling</code>	integer	(50 - 200) as percent.
<code>invert texture</code>	boolean	(default: <code>false</code>)
<code>texture kind</code>	Valid values: blocks canvas frosted texture document tiny lens	The type of texture to use.
<code>texture definition</code>	alias	The file that contains the texture image.

The following code sample demonstrates how to apply the Glass filter using the `filter` command of the `glass` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using glass filter -
  with options {class:glass, distortion:20, smoothness:100-
    scaling:155, texture kind:tiny lens, texture definition:"Hard Disk:Photoshop
    CS2:Textures:Texture1"}
```

gray color

Options that can be specified for a Gray color: the numeric values of component colors. The `Gray color` class inherits properties from the [color value](#) class.

This class a property of the [color value](#) object and is used to define a record that specifies options when creating color value.

Properties

Property	Value Type	What it is
<code>gray value</code>	real	The gray value (0.0 - 100.0; default: 0.0).

high pass

Options that may be supplied when applying a High Pass filter.

The `high pass` class supplies properties for the [filter options](#) class. The `high pass` class is used only to define a record whose properties specify options when applying a High Pass filter. `high pass` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `high pass` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>radius</code>	real	(0.1 - 250.0) in pixels.

The following code sample demonstrates how to apply the High Pass filter using the `filter` command of the `high pass` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using high pass filter -
  with options {class:high pass, radius:25.5}
```

history state, history states

A history state for the document.

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	All of this objects properties returned in a single record.
container	reference	Read-only. The object's container.
name	Unicode text	The history state's name.
snapshot	boolean	Read-only. Indication of whether to the history state is a snapshot.

Valid Commands

[count](#)
[exists](#)
[index](#)

HSB color

Options that can be specified for an HSB color: the numeric values of component colors. The `HSB color` class inherits properties from the [color value](#) class.

This class a property of the [color value](#) object and is used to define a record that specifies options when creating color value.

Properties

Property	Value Type	What it is
<code>hue</code>	real	The hue value (0.0 - 100.0; default: 0.0).
<code>saturation</code>	real	The saturation value (0.0 - 100.0; default: 0.0).
<code>brightness</code>	real	The brightness value (0.0 - 100.0; default: 0.0).

Illustrator paths export options

Options that can be specified when exporting a path item to an Adobe Illustrator® file. See ['path item, path items' on page 104](#).

Properties

Property	Value Type	What it is
<code>path name</code>	Unicode text	The name of path to export. Note: Valid only when <code>target path:</code> <code>named path</code> .
<code>target path</code>	Valid values: all paths document bounds named path	The sort of path(s) to export (default: <code>document bounds</code>).

indexed mode options

Options that can be specified when changing the document mode to Indexed using the `change mode` command. The `indexed mode` class inherits properties from the [change mode options](#) class.

This class is used to define a record containing properties used to specify options when changing a document's mode. `Indexed mode options` can only be supplied in conjunction with the `change mode` command. It is not possible to get or create a `Indexed mode options` object. See the [change mode](#) command for more details.

Properties

Property	Value Type	What it is
<code>colors in palette</code>	integer	The number of colors in the Color palette. Note: Valid only when palette: local adaptive; palette: local perceptual; palette: local selective; palette: Mac OS system; palette: uniform; palette: web; OR palette: Windows system. See palette .
<code>dither</code>	Valid values: diffusion noise none pattern	The dither type to use.
<code>dither amount</code>	integer	The amount of dither to use (1 - 100). Note: Valid only when <code>dither: diffusion</code> .
<code>forced colors</code>	Valid values: black and white none primaries web	The type of colors to force into the color palette.
<code>matte background color</code>	Valid values: matte black matte foreground color matte Netscape gray none semi gray white matte	The color to use to fill antialiased edges adjacent to transparent areas of the image (default: <code>matte white</code>). Note: When <code>transparency: false</code> , the matte color is applied to transparent areas. See transparency .

Property	Value Type	What it is (Continued)
<code>palette</code>	Valid values: exact local adaptive local perceptual local selective Mac OS system master adaptive master perceptual master selective previous uniform web Windows system	The type of palette to use (default: <code>local selective</code>).
<code>preserve exact colors</code>	boolean	Indication of whether to protect colors in the image that contain entries in the color table from being dithered. Note: Valid only when <code>dither: diffusion</code> . See dither .
<code>transparency</code>	boolean	Indication of whether to preserve transparent areas of the image during conversion to GIF format.

info-object

Document metadata. This class is a property of the `document` object and is used to define a record containing properties that specify application preferences. See [‘document, documents’ on page 47](#).

Properties

Property	Value Type	What it is
<code>best type</code>	type class	Read-only. The best type for the object's value.
<code>class type</code>	class	The object's class.
<code>default type</code>	type class	Read-only. The default type for the object's value.
<code>properties</code>	record	All of this object's properties returned in a single record.
<code>author</code>	Unicode text	
<code>author position</code>	Unicode text	
<code>caption</code>	Unicode text	
<code>caption writer</code>	Unicode text	
<code>category</code>	Unicode text	
<code>city</code>	Unicode text	
<code>container</code>	reference	Read-only. The object's container.
<code>copyright notice</code>	Unicode text	
<code>copyrighted</code>	Valid values: copyrighted work public domain unmarked	The document's copyrighted status.
<code>country</code>	Unicode text	
<code>creation date</code>	Unicode text	
<code>credit</code>	Unicode text	
<code>EXIF</code>	list of lists: list (list {tag, tag data}), ...	Read-only. Camera data that includes camera settings used when the image was taken. Sample list values are: tag = "camera"; tag value = "Canon".
<code>headline</code>	Unicode text	
<code>instructions</code>	Unicode text	
<code>job name</code>	Unicode text	
<code>keywords</code>	list of Unicode text	A list of keywords that can identify the document or its contents.
<code>owner url</code>	Unicode text	

Property	Value Type	What it is (Continued)
province or state	Unicode text	
source	Unicode text	
supplemental categories	list of Unicode text	
title	Unicode text	
transmission reference	Unicode text	
urgency	Valid values: four high low none normal seven six three two	

inversion

The `inversion` class supplies options for the [adjustment options](#) class to invert the colors in an image.

The `inversion` class, which is analogous to the Invert command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply an Invert adjustment to an image. The `inversion` class can be used only in conjunction with the `adjust` command. It is not possible to get or create an `inversion` object.

See the [adjust](#) command for additional details.

The following code sample demonstrates how to apply a Inversion adjustment using the `adjust` command of the `inversion` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using inversion adjustment-
  with options {class:inversion}
```

JPEG save options

Options that can be specified when saving a document in JPEG format. See the [save](#) command for additional details. The `JPEG save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a JPEG file. `JPEG save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `JPEG save options` object.

Properties

Property	Value Type	What it is
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>format options</code>	Valid values: optimized progressive standard	The download format to use (default: standard).
<code>matte background color</code>	Valid values: matte black matte foreground color matte Netscape gray none semi gray white matte	The color to use to fill antialiased edges adjacent to transparent areas of the image (default: white matte). Note: When <code>transparency: false</code> , the matte color is applied to transparent areas. See 'transparency' on page 77 .
<code>quality</code>	integer	The quality of the produced image (0 - 12; default: 3).
<code>scans</code>	integer	The number of increasingly detailed scans to use to display the image on the screen (3 - 5). Note: Valid only when <code>format options: progressive</code> . See format options .

The following code creates the `myOptions` variable to hold the `JPEG save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document
  set myOptions to {class:JPEG save options, -
    embed color profile: true, format options: progressive, scans: 3}
  save current document in file myFile as JPEG with options -
    myOptions appending no extension without copying
end tell
```

Lab color

Options that can be specified for an Lab color: numeric values for the component colors. The `Lab color` class inherits properties from the [color value](#) class.

This class a property of the [color value](#) object and is used to define a record that specifies options when creating color value.

See the [convert color](#) command for additional details.

Properties

Property	Value Type	What it is
<code>value_L</code>	real	The L value (0.0 - 100.0).
<code>value_a</code>	real	The a value (-128.0 - 127.0).
<code>value_b</code>	real	The b value (-128.0 - 127.0).

layer, layers

A layer object. The following classes inherit properties from the `layer` class: [art layer](#), [art layers](#), [layer set](#), [layer sets](#).

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	Read-only. All of this object's properties returned in a single record.
all locked	boolean	Read-only. Indication of whether all of the layer's contents are locked or editable.
blend mode	Valid Values: color blend color burn color dodge darken difference dissolve exclusion hard light hue blend lighten linear burn linear dodge linear light luminosity multiply normal overlay pass through pin light saturation blend screen soft light vivid light	The mode to use when compositing an object.
bounds	list	Read-only. A list of coordinates that describe the corners of the bounding rectangle of the layer.
container	reference	Read-only. The object's container.
linked layers	list of layers	Read-only. The layers linked to this layer.
name	Unicode text	The layer's name.
opacity	real	The layer's master opacity (as percent) (0.0 - 100.0).
visible	boolean	Indication of whether the layer is visible.

Valid Commands

[delete](#)

[index](#)

[link](#)

[make](#)

[rotate](#)

[scale](#)

[translate](#)

[unlink](#)

layer comp, layer comps

A "snapshot" of a temporary composition of the layers in a document.

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	Read-only. All of this object's properties returned in a single record.
appearance	boolean	Indication of whether to use layer appearance (layer styles) settings.
comment	anything	Read-only. A description of the layer comp.
container	reference	Read-only. The object's container.
name	Unicode text	The layer's name.
position	boolean	Indication of whether to use layer position.
selected	boolean	Indication of whether the layer comp is selected.
visibility	boolean	Indication of whether to use layer visibility settings.

Valid Commands

[apply](#)
[delete](#)
[index](#)
[make](#)
[recapture](#)
[reset from comp](#)

layer set, layer sets

A layer set.

Elements

Element	Refer to by
art layer	name, numeric index, as a range of elements, before after another element, satisfying a test.
layer	name, numeric index, as a range of elements, before after another element, satisfying a test.
layer set	name, numeric index, as a range of elements, before after another element, satisfying a test.

Properties

Property	Value Type	What it is
best type	type class	Read-only. (Inherited from the <code>layer</code> class) The best type for the object's value.
class type	class	Read-only. (Inherited from the <code>layer</code> class) The object's class.
default type	type class	Read-only. (Inherited from the <code>layer</code> class) The default type for the object's value.
index	integer	Read-only. (Inherited from the <code>layer</code> class) The index of this instance of the object.
properties	record	Read-only. (Inherited from the <code>layer</code> class) All of this object's properties returned in a single record.
all locked	boolean	Read-only. (Inherited from the <code>layer</code> class) Indicates whether the contents in the layers contained in the layer set are editable.

Property	Value Type	What it is
blend mode	Valid Values: color blend color burn color dodge darken difference dissolve exclusion hard light hue blend lighten linear burn linear dodge linear light luminosity multiply normal overlay pass through pin light saturation blend screen soft light vivid light	(Inherited from the <code>layer</code> class) The mode to use when compositing an object.
bounds	list	Read-only. (Inherited from the <code>layer</code> class) The bounding rectangle of the layer.
container	reference	Read-only. (Inherited from the <code>layer</code> class) The object's container.
linked layers	list of layers	Read-only. (Inherited from the <code>layer</code> class) The layers linked to this layer set.
name	Unicode text	(Inherited from the <code>layer</code> class) The layer set's name.
opacity	real	(Inherited from the <code>layer</code> class) The layer set's master opacity (0.0 - 100.0).
visible	boolean	(Inherited from the <code>layer</code> class) Indication of whether the layer set is visible.
enabled channels	list of channels	Channels that are enabled for the layer set. Note: Must be a list of component channels.

Valid Commands

[delete](#)
[duplicate](#)
[exists](#)
[index](#)
[link](#)
[make](#)
[merge](#)
[move](#)
[rotate](#)
[scale](#)
[translate](#)

[unlink](#)

lens blur

Options that can be specified when applying a Lens Blur filter.

The `lens blur` class supplies properties for the [filter options](#) class. The `lens blur` class is used only to define a record whose properties specify options when applying a Lens Blur filter. `lens blur` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `lens blur` object.

See the [filter](#) command for additional details.

Properties

Property	Value type	What it is
<code>amount</code>	number (long)	The amount of noise (default: 0).
<code>blade curvature</code>	number (long)	The blade curvature of the iris (default: 0).
<code>brightness</code>	number (long)	The brightness for the specular highlights (default: 0).
<code>distribution</code>	Valid values: gaussian uniform	The distribution value for the noise (default: uniform).
<code>focal distance</code>	number (long)	The blur focal distance for the depth map (default: 0).
<code>invert depth map</code>	boolean	Indicates whether the depth map is inverted (default: false).
<code>monochromatic</code>	boolean	Indicates whether the noise is monochromatic (default: false).
<code>radius</code>	number (long)	The radius of the iris (default: 15).
<code>rotation</code>	number (long)	The rotation of the iris (default: 0).
<code>shape</code>	Valid values: heptagon hexagon octagon pentagon square triangle	The shape of the iris (default: hexagon).
<code>source</code>	Valid values: image highlight layer mask none transparency channel	The source for the depth map (default: none).
<code>threshold</code>	number (long)	The threshold for the specular highlights (default: 0).

The following code sample demonstrates how to apply the Lens Blur filter using the `filter` command of the `lens blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using lens blur filter ↵
  with options {class:lens blur, radius:13}
```

lens flare

Options that may be supplied when applying a Lens Flare filter.

The `lens flare` class supplies properties for the [filter options](#) class. The `lens flare` class is used only to define a record whose properties specify options when applying a Lens Flare filter. `lens flare` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `lens flare` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>brightness</code>	integer	(10 - 300) as percent.
<code>flare center</code>	list	The x and y coordinates of the center of the flare (unit value).
<code>lens type</code>	Valid values: Movie Prime Prime 105 Prime 35 zoom	

The following code sample demonstrates how to apply the Lens Flare filter using the `filter` command of the `lens flare` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using lens flare filter ~
  with options {class:lens flare, brightness:209, flare center: {2 as inches, ~
    4 as inches}, lens type:Prime 35}
```

levels adjustment

The `levels adjustment` class supplies options for the [adjustment options](#) class to correct the tonal range and color balance of an image.

The `levels adjustment` class, which is analogous to the Levels command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when applying a Levels adjustment to an image. The `levels adjustment` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `levels adjustment` object.

See the [adjust](#) command for additional details.

Properties

Property	Value Type	What it is
<code>input range start</code>	integer	(0 - 253)
<code>input range end</code>	integer	$((inputRangeStart + 2) - 255)$
<code>input range gamma</code>	real	(0.10 - 9.99)
<code>output range start</code>	integer	(0 - 253)
<code>output range end</code>	integer	$((outputRangeStart + 2) - 255)$

The following code sample demonstrates how to apply a Levels Adjustment using the `adjust` command of the `levels adjustment` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using levels adjustment-
with options {class:levels adjustment, input range start:4, -
input range end:245, input range gamma:4.8, output range start:5, -
output range end: 7}
```

maximum filter

Options that may be supplied when applying a Maximum filter.

The `maximum filter` class supplies properties for the [filter options](#) class. The `maximum filter` class is used only to define a record whose properties specify options when applying a Maximum filter. `maximum filter` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `maximum filter` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>radius</code>	real	(1 - 100) in pixels.

The following code sample demonstrates how to apply the Maximum filter using the `filter` command of the `maximum filter` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using maximum filter -
    with options {class:maximum filter, radius:29}
```

median noise

Options that may be supplied when applying a Median Noise filter.

The `median noise` class supplies properties for the [filter options](#) class. The `median noise` class is used only to define a record whose properties specify options when applying a Median Noise filter. `median noise` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `median noise` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>radius</code>	real	(1 - 100) in pixels.

The following code sample demonstrates how to apply the Median Noise filter using the `filter` command of the `median noise` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using median noise filter -
  with options {class:median noise, radius:22}
```

minimum filter

Options that may be supplied when applying a Minimum filter.

The `minimum filter` class supplies properties for the [filter options](#) class. The `minimum filter` class is used only to define a record whose properties specify options when applying a Minimum filter. `minimum filter` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `minimum filter` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>radius</code>	real	(1 - 100) in pixels.

The following code sample demonstrates how to apply the Minimum filter using the `filter` command of the `minimum filter` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using minimum filter -
  with options {class:minimum filter, radius:29}
```

mix channels

The `mix channels` class supplies options for the [adjustment options](#) class to modify a targeted (output) color channel using a mix of the existing color channels in the image.

The `mix channels` class, which is analogous to the Channel Mixer command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when applying a Channel Mixer adjustment to an image. The `mix channels` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `mix channels` object.

See the [adjust](#) command for additional details.

Note: Valid only for RGB or CMYK documents.

Properties

Property	Value Type	What it is
<code>output channels</code>	list of anything	<p>A list of channel specifications. For each component channel, you must specify a list of adjustment values (-200 - 200) followed by a 'constant' value (-200 - 200).</p> <p>Note: When <code>monochrome mixing: true</code>, the maximum number of channel value specifications is 1.</p> <p>Note: Valid only when the document mode is RGB or CMYK. See <code>mode</code> in the Properties table of the document, documents class.</p>
<code>monochrome mixing</code>	boolean	Indication of whether to use monochrome mixing (default: <code>false</code>).

The following code sample demonstrates how to apply a Mix Channels adjustment using the `adjust` command of the `mix channels` class (inherited from the `adjustment options` super class). This code assumes an RGB document mode and uses the default value for the `monochrome mixing` property by not specifying a value.

```
...
adjust current layer of the current document using mix channels adjustment-
  with options {class:mix channels, output channels:{red:{-42, 62, 5, 116},
green:{0, 172, 136, 0}, blue: {-66, 0, 100, -112}}}
```

motion blur

Options that may be supplied when applying a Motion Blur filter.

The `motion blur` class supplies properties for the [filter options](#) class. The `motion blur` class is used only to define a record whose properties specify options when applying a Motion Blur filter. `motion blur` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `motion blur` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>angle</code>	integer	(angle: -360 - 360)
<code>radius</code>	real	(radius: 1 - 999) in pixels.

The following code sample demonstrates how to apply the Motion Blur filter using the `filter` command of the `motion blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using motion blur filter -
  with options {class:motion blur, angle: -224, radius:29}
```

no color

An object that represents a missing color. The `no color` class inherits properties from the [color value](#) class.

notifier, notifiers

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

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.

Properties

Property	Value type	What it is
<code>event</code>	string	Read-only. The event ID in four characters or a unique string that the notifier is associated with.
<code>event class</code>	string	Read-only. The class ID of the event.
<code>event file</code>	file	Read-only. The path to the file to execute when the event occurs/activates the notifier.
<code>typename</code>	string	Read-only. The class name of the referenced <code>notifier</code> object.

Valid commands:

[delete](#)
[exists](#)
[index](#)
[make](#)

NTSC colors

Options that may be supplied when applying the NTSC colors filter.

The `NTSC colors` class supplies properties for the [filter options](#) class. The `NTSC colors` class is used only to define a record whose properties specify options when using the `filter` command to apply an NTSC colors filter. `NTSC colors` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `NTSC colors` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the NTSC colors filter using the `filter` command of the `motion blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using NTSC colors filter ~
  with options {class:NTSC colors}
```

ocean ripple

Options that may be supplied when applying an Ocean Ripple filter.

The `ocean ripple` class supplies properties for the [filter options](#) class. The `ocean ripple` class is used only to define a record whose properties specify options when applying an Ocean Ripple filter. `ocean ripple` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create an `ocean ripple` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>ripple size</code>	integer	(1 - 15)
<code>ripple magnitude</code>	integer	(0 - 20)

The following code sample demonstrates how to apply the Ocean Ripple filter using the `filter` command of the `ocean ripple` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using ocean ripple filter ~
  with options {class:ocean ripple, ripple size: 4, ripple magnitude:2}
```

offset filter

Options that may be supplied when applying an Offset filter.

The `offset filter` class supplies properties for the [filter options](#) class. The `offset filter` class is used only to define a record whose properties specify options when applying an Offset filter. `offset filter options` can be supplied only in conjunction with the `filter` command. It is not possible to get or create an `offset filter` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>horizontal offset</code>	real	(unit value) (min/max amounts depend on layer size).
<code>vertical offset</code>	real	(unit value) (min/max amounts depend on layer size).
<code>undefined areas</code>	Valid values: repeat edge pixels set to layer fill wrap around	The method to use to fill areas left undefined when the image is moved.

The following code sample demonstrates how to apply the Offset filter using the `filter` command of the `offset filter` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using offset filter ~
  with options {class:offset filter, horizontal offset: 3 cm, ~
    vertical offset: 2 cm, undefined areas:wrap around}
```

open options

Options used with the open command.

The following classes inherit properties from the open options class: [camera raw open options](#), [EPS open options](#), [PDF open options](#), [Photo CD open options](#), [raw format open options](#).

Valid Commands

[open](#)

path item, path items

An artwork path.

Elements

Element	Refer to by
sub path item	numeric index, as a range of elements, before/after another element, satisfying a test

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	Read-only. All of this object's properties returned in a single record.
container	reference	Read-only. The object's container.
entire path	list of sub path info objects	Read-only. The item's sub paths.
kind	Valid values: clipping normal text mask vector mask work	The type of path.
name	Unicode text	The path item's name.

The following code creates a path in three segments that end up looking like an ice cream cone: two connected diagonal lines that form a V (or the "cone"), and a curved line that arcs above the V to create a 2-d ice cream cone.

```
tell application "Adobe Photoshop CS2"
    set ruler units of settings to pixel units
    set type units of settings to pixel units

    set docRef to make new document with properties {height:700, width:500,-
        name:"Snow Cone"}

    set pathPointInfo1 to {class:path point info, kind:corner point,-
        anchor:{100, 100}, left direction:{100, 100}, right direction:{100, 100}}
    set pathPointInfo2 to {class:path point info, kind:corner point,-
        anchor:{150, 200}, left direction:{150, 200}, right direction:{150, 200}}
    set subPathInfo1 to {class:sub path info, entire sub path:{pathPointInfo1,-
        pathPointInfo2}, operation:shape xor, closed:false}
```

```
set pathPointInfo3 to {class:path point info, kind:corner point, anchor:-
    {150, 200}, left direction:{150, 200}, right direction:{150, 200}}
set pathPointInfo4 to {class:path point info, kind:corner point, anchor:-
    {200, 100}, left direction:{200, 100}, right direction:{200, 100}}
set subPathInfo2 to {class:sub path info, entire sub path:{pathPointInfo3,-
    pathPointInfo4}, operation:shape xor, closed:false}

set pathPointInfo5 to {class:path point info, kind:corner point, anchor:-
    {200, 100}, left direction:{200, 100}, right direction:{200, 100}}
set pathPointInfo6 to {class:path point info, kind:corner point, anchor:-
    {150, 50}, left direction:{100, 50}, right direction:{200, 50}}
set pathPointInfo7 to {class:path point info, kind:corner point, anchor:-
    {100, 100}, left direction:{100, 100}, right direction:{100, 100}}
set subPathInfo3 to {class:sub path info, entire sub path:{pathPointInfo5,-
    pathPointInfo6, pathPointInfo7}, operation:shape xor, closed:false}

set newPathItem to make new path item in docRef with properties {entire path:-
    {subPathInfo1, subPathInfo2, subPathInfo3}, name:"Snow Cone", kind:normal}

end tell
```

Valid Commands

[create selection](#)
[delete](#)
[deselect](#)
[duplicate](#)
[exists](#)
[fill path](#)
[index](#)
[make](#)
[make clipping path](#)
[select](#)
[stroke path](#)

path point, path points

A list of [sub_path_info](#) objects.

Tip: You do not use the path point object to create a path. Rather, you use the path point object to retrieve information about a path. To create points that describe path segments, see [path_point_info](#).

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
properties	record	Read-only. All of this object's properties returned in a single record.
anchor	list	Read-only. The position (coordinates) of the anchor point.
container	reference	Read-only. The object's container.
entire path	list of sub_path_info objects	Read-only. The item's sub paths.
kind	Valid values: corner point smooth point	The type of point.
left direction	list	Read-only. The location of the left-direction ('in' position) point.
right direction	list	Read-only. The location of the right-direction ('out' position) point.

Valid Commands

[exists](#)
[index](#)

path point info

A point on a path, expressed in three sets of coordinates:

- anchor point
- left direction point
- 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.

Tip: The entire `path` property of the path item class returns `path point info` objects. See the Properties table for the [path item](#), [path items](#) object.

Properties

Property	Value Type	What it is
<code>anchor</code>	list of coordinates	Read-only. The position (in x and y coordinates) of the anchor point.
<code>kind</code>	Valid values: corner point smooth point	The type of point.
<code>left direction</code>	list of coordinates	Read-only. The location (in x and y coordinates) of the left-direction ('in' position) point.
<code>right direction</code>	list of coordinates	Read-only. The location (in x and y coordinates) of the right-direction ('out' position) point.

Valid Commands

[exists](#)

PDF open options

Settings related to opening a generic PDF document. The `PDF open options` class inherits properties from the [open options](#) class.

This class is used to define a record containing properties that specify options when opening a document as a PDF file. `PDF open options` can be supplied only in conjunction with the `open` command. It is not possible to get or create a `PDF open options` object.

Properties

Property	Value Type	What it is
<code>constrain proportions</code>	boolean	Indication of whether to constrain the proportions of the original image.
<code>bits per channel</code>	Valid values: eight one sixteen thirtytwo	The number of bits per channel.
<code>constrain proportions</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>crop page</code>	Valid values: art box bleed box bounding box crop box media box trim box	The method of cropping to use.
<code>height</code>	UnitValue	Deprecated for Adobe Photoshop CS2.
<code>mode</code>	Valid values: CMYK grayscale Lab RGB	The color model to use.
<code>name</code>	string	The name of the document.
<code>page</code>	integer	The page to which to open the document.
<code>resolution</code>	real	The resolution of the document (in pixels per inch).
<code>suppress warnings</code>	boolean	Indication of whether to suppress warnings when opening the document.
<code>use antialias</code>	boolean	Indication of whether to use antialias.
<code>use page number</code>	boolean	Indication of whether the value specified in the <code>page</code> property will refer to an image number when <code>usePageNumber = false</code> . See page .
<code>width</code>	unit value	Deprecated for Adobe Photoshop CS2.

The following code opens a PDF document.

```
tell application "Adobe Photoshop CS2"
  set myFilePath to alias "Applications:PDFFiles:MyFile.pdf"
  open myFilePath as PDF with options ¬
    {class:PDF open options, height:pixels 100, ¬
      width:pixels 200, mode:RGB, resolution:72, ¬
      use antialias:true, page:3, ¬
      constrain proportions:false}
end tell
```

PDF save options

Options that can be specified when saving a document in PDF format. See the [save](#) command for additional details. The `PDF save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a PDF file. `PDF save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `PDF save options` object.

Properties

Property	Value Type	What it is
<code>color conversion</code>	boolean	Indication of whether to convert the color profile to a destination profile.
<code>convert to right bit</code>	boolean	Indication of whether to convert a 16-bit image to 8-bit for better compatibility with other applications.
<code>descripton</code>	string	Description of the save options to use.
<code>destination profile</code>	string	Description of the final RGB or CMYK output device, such as a monitor or a press standard.
<code>downgrade color profile</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>down sample</code>	Valid values: none PDF average PDF bicubic PDF subsample	The down sample method to use.
<code>down sample size</code>	number (double)	The size to downsample images if they exceed the limit in pixels per inch.
<code>downSampleSizeLimit</code>	number (double)	Limits downsampling or subsampling to images that exceed this value in pixels per inch.
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>embed fonts</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>embed thumbnail</code>	boolean	Indication of whether to include a small preview image in Adobe PDF files.

Property	Value Type	What it is (Continued)
<code>encoding</code>	Valid values: JPEG JPEG 2000 high JPEG 2000 lossless JPEG 2000 low JPEG 2000 med JPEG 2000 med high JPEG 2000 med low JPEG high JPEG low JPEG med JPEG med high JPEG med low none ZIP ZIP 4	The encoding or compression method to use (default: ZIP).
<code>image interpolation</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>JPEG quality</code>	integer	The quality of the produced image (0 - 12; default: 10). Note: Valid only when <code>encoding</code> : JPEG. See encoding .
<code>optimize for web</code>	boolean	Indication of whether to improve performance of PDF files on Web servers.
<code>output condition</code>	string	An optional comment field for inserting descriptions of the output condition. The text is stored in the PDF/X file.
<code>output condition ID</code>	string	Identifier for the output condition.
<code>PDF compatibility</code>	Valid values: PDF 13 PDF 14 PDF 15 PDF 16	The PDF version to make the document compatible with.
<code>PDF standard</code>	Valid values: none PDFX1a2001 standard PDFX1a2003 standard PDFX32002 standard PDFX32003 standard	The PDF standard to make the document compatible with.
<code>preserve editing</code>	boolean	Indication of whether to reopen the PDF in Adobe Photoshop CS2 with native Photoshop data intact.
<code>preset file</code>	string	The preset file to use for settings. Note: This option overrides other settings.
<code>profile inclusion policy</code>	boolean	Indication of whether to show which profiles to include.

Property	Value Type	What it is (Continued)
<code>registry name</code>	string	URL where the output condition is registered.
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.
<code>save annotations</code>	boolean	Indication of whether to save the annotations.
<code>save spot colors</code>	boolean	Indication of whether to save the spot colors.
<code>transparency</code>	boolean	Indication of whether to use transparency (default: <code>true</code>).
<code>tile size</code>	number (long)	Compression option. Note: Valid only when <code>encoding: JPEG 2000</code> .
<code>transparency</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>use outlines for text</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>vector data</code>	boolean	Deprecated for Adobe Photoshop CS2.
<code>view</code>	boolean	Indication of whether to open the saved PDF in Adobe Acrobat.

The following code creates the `myOptions` variable to hold the PDF save options, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
    set theDocRef to make new document

    --create a variable that contains a color object of the RGB color class
    --whose color is red
    set theTextColor to {class:RGB color, red:255, green:0, blue:0}

    -- create a text layer
    set theTextLayer to make new art layer in theDocRef with-
        properties {kind:text layer}

    --Set the contents, size, position and color of the text layer
    set contents of text object of theTextLayer to "Hello, World!"
    set size of text object of theTextLayer to 36
    set position of text object of theTextLayer to {0.75, 1}
    set stroke color of text object of theTextLayer to theTextColor
    set myOptions to {class:PDF save options, -
        embed color profile:true, save alpha channels:true,-
        vector data:true, use outlines for text:true}
    save current document in file myFile as Photoshop PDF with options -
        myOptions appending no extension without copying
end tell
```

Photo CD open options

Options that can be specified when opening a Kodak Photo CD (PCD) files, including high-resolution files from Pro Photo CD discs. The `Photo CD open options` class inherits properties from the [open options](#) class.

This class is used to define a record containing properties that specify options when opening a Photo CD document. `Photo CD open options` can be supplied only in conjunction with the `open` command. It is not possible to get or create a `Photo CD open options` object.

Properties

Property	Value Type	What it is
<code>color profile name</code>	Unicode text	The profile to use when reading the image.
<code>color space</code>	Valid Values: Lab 16 Lab 8 RGB 16 RGB 8	The colorspace for the image.
<code>orientation</code>	Valid Values: landscape portrait	The image orientation.
<code>pixel size</code>	Valid Values: extra large large maximum medium minimum small	The image dimensions.
<code>resolution</code>	real	The image resolution (in pixels per inch).

The following code opens a Photo CD document.

```
tell application "Adobe Photoshop CS2"
  set myFilePath to alias "Applications:PhCDFiles:MyFile.pcd"
  open myFilePath as PDF with options -
    {class:Photo CD open options, color profile name:myColorProfile, -
    color space:RGB 8, resolution:72, -
    orientation:landscape, pixel size:medium}
end tell
```

Photoshop DCS 1.0 save options

Options that can be specified when saving a document in Photoshop DCS 1.0 format. See the [save](#) command for additional details. The `Photoshop DCS 1.0 save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a Photoshop DCS 1.0 file. `Photoshop DCS 1.0 save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `Photoshop DCS 1.0 save options` object.

Properties

Property	Value Type	What it is
<code>DCS color</code>	Valid Values: color composite grayscale composite no composite PostScript	The DCS color to use (default: <code>color composite</code>).
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>embed fonts</code>	boolean	Indication of whether to embed fonts (default: <code>false</code>). Note: Valid only when <code>vector data: true</code> . See vector data .
<code>encoding</code>	Valid Values: ASCII binary high quality JPEG low quality JPEG maximum quality JPEG medium quality JPEG	The type of encoding to use (default: <code>binary</code>).
<code>halftone screen</code>	boolean	Indication of whether to include the halftone screen (default: <code>false</code>).
<code>image interpolation</code>	boolean	Indication of whether to use image interpolation (default: <code>false</code>).
<code>preview type</code>	Valid Values: eight bit Mac OS eight bit TIFF JPEG Mac OS monochrome Mac OS monochrome TIFF none	The type of preview (default: <code>eight bit Mac OS</code>).
<code>transfer function</code>	boolean	Indication of whether to include the Transfer functions to compensate for dot gain between the image and film (default: <code>false</code>).
<code>vector data</code>	boolean	Indication of whether to include vector data (un-rasterized text) (default: <code>false</code>).

The following code creates the `myOptions` variable to hold the Photoshop DCS 1.0 save options, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  -- create a text layer
  set theTextLayer to make new art layer in theDocRef with-
    properties {kind:text layer}

  --Set the contents, size, position and color of the text layer
  set contents of text object of theTextLayer to "Hello, World!"
  set size of text object of theTextLayer to 36
  set position of text object of theTextLayer to {0.75, 1}
  set stroke color of text object of theTextLayer to theTextColor
  set myOptions to {class:Photoshop DCS 1.0 save options, -
    embed color profile:true, save alpha channels:true,-
    vector data:true}
  save current document in file myFile as Photoshop DCS 1.0 with options -
    myOptions appending no extension without copying
end tell
```

Photoshop DCS 2.0 save options

Options that can be specified when saving a document in Photoshop DCS 2.0 format. See the [save](#) command for additional details. The `Photoshop DCS 2.0 save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a Photoshop DCS 2.0 file. `Photoshop DCS 2.0 save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `Photoshop DCS 2.0 save options` object.

Properties

Property	Value Type	What it is
<code>DCS color</code>	Valid Values: color composite grayscale composite no composite PostScript	The DCS color to use (default: no composite PostScript).
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>embed fonts</code>	boolean	Indication of whether to embed fonts (default: false). Note: Valid only if a text layer is included
<code>encoding</code>	Valid Values: ASCII binary high quality JPEG low quality JPEG maximum quality JPEG medium quality JPEG	The type of encoding to use (default: binary)
<code>halftone screen</code>	boolean	Indication of whether to include the halftone screen (default: false).
<code>image interpolation</code>	boolean	Indication of whether to use image interpolation (default: false).
<code>multifile DCS</code>	boolean	Indication of whether to save color channels as multiple files or a single file (default: false).
<code>preview type</code>	Valid Values: eight bit Mac OS eight bit TIFF JPEG Mac OS monochrome Mac OS monochrome TIFF none	The type of preview (default: eight bit Mac OS).
<code>save spot colors</code>	boolean	Indication of whether to save the spot colors.

Property	Value Type	What it is (Continued)
transfer function	boolean	Indication of whether to include the Transfer functions to compensate for dot gain between the image and film (default: <code>false</code>).
vector data	boolean	Indication of whether to include vector data (un-rasterized text) (default: <code>false</code>).

The following code creates the `myOptions` variable to hold the Photoshop DCS 2.0 save options, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  -- create a text layer
  set theTextLayer to make new art layer in theDocRef with~
    properties {kind:text layer}

  --Set the contents, size, position and color of the text layer
  set contents of text object of theTextLayer to "Hello, World!"
  set size of text object of theTextLayer to 36
  set position of text object of theTextLayer to {0.75, 1}
  set stroke color of text object of theTextLayer to theTextColor
  set myOptions to {class:Photoshop DCS 2.0 save options, ~
    embed color profile:true, save spot colors:true, ~
    vector data:true}
  save current document in file myFile as Photoshop DCS 2.0 with options ~
    myOptions appending no extension without copying
end tell
```

Photoshop save options

Options that can be specified when saving a document in PSD format. See the [save](#) command for additional details. The `PSD save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a Photoshop file. `PSD save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `PSD save options` object.

Properties

Property	Value Type	What it is
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.
<code>save annotations</code>	boolean	Indication of whether to save the annotations.
<code>save layers</code>	boolean	Indication of whether to save the layers.
<code>save spot colors</code>	boolean	Indication of whether to save the spot colors.

The following code creates the `myOptions` variable to hold the `Photoshop save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
    make new document

    -- create a text layer
    set theTextLayer to make new art layer in theDocRef with-
        properties {kind:text layer}

    --Set the contents, size, position and color of the text layer
    set contents of text object of theTextLayer to "Hello, World!"
    set size of text object of theTextLayer to 36
    set position of text object of theTextLayer to {0.75, 1}
    set stroke color of text object of theTextLayer to theTextColor
    set myOptions to {class:Photoshop save options, -
        embed color profile:true, save spot colors:true,-
        vector data:true}
    save current document in file myFile as Photoshop format with options -
        myOptions appending no extension without copying
end tell
```

PICT file save options

Options that can be specified when saving a document in PICT format. See the [save](#) command for additional details. The `PICT file save options` class inherits properties from the [save options](#) class. This class is used to define a record containing properties used to specify options when saving a document as a PICT file. `PICT file save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `PICT file save options` object.

Properties

Property	Value Type	What it is
<code>compression</code>	Valid Values: high quality JPEG low quality JPEG maximum quality JPEG medium quality JPEG none	(default: none).
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>resolution</code>	Valid Values: eight four sixteen thirty two two	The number of bits per pixel.
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.

The following code creates the `myOptions` variable to hold the `PICT file save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  set myOptions to {class:PICT file save options, ~
    embed color profile: true, save alpha channels: true, ~
    resolution:sixteen}
  save current document in file myFile as PICT file format with options ~
    myOptions appending no extension without copying
end tell
```

PICT resource save options

Options that can be specified when saving a document as a PICT resource. See the [save](#) command for additional details. The `PICT resource save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a PICT resource file. `PICT resource save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `PICT resource save options` object.

Properties

Property	Value Type	What it is
<code>compression</code>	Valid Values: high quality JPEG low quality JPEG maximum quality JPEG medium quality JPEG none	(default: none).
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>name</code>	Unicode text	The name of the PICT resource.
<code>resolution</code>	Valid Values: eight four sixteen thirty two two	The number of bits per pixel.
<code>resource id</code>	integer	The ID of the PICT resource (default: 128).
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.

The following code creates the `myOptions` variable to hold the `PICT resource save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
    make new document

    set myOptions to {class:PICT resource save options, -
        embed color profile: true, save alpha channels: true, -
        name:"My PICT resource", resolution:sixteen}
    save current document in file myFile as PICT resource format with options -
        myOptions appending no extension without copying
end tell
```

picture package options

Options that can be specified for a picture package.

Property	Value type	What it is
content	Valid values: caption copyright credit filename none origin user	The content information (default: <code>user</code>).
flatten	boolean	Indicates whether all layers in the final document are flattened (default: <code>true</code>).
font	Valid values: arial courier new helvetica times new roman	The font used for security text (default: <code>GalleryFontType.ARIAL</code>).
fontSize	number (long)	The font size used for security text (default: 12).
layout	string	The layout to use to generate the picture package (default: "(2) 5x7").
mode	Valid values: bitmap CMYK grayscale lab RGB	The color profile to use as the document mode (default: <code>RGB</code>).
opacity	number (long)	The web page security opacity as a percent (default: 100).
resolution	number (double)	The resolution of the document in pixels per inch (default: 72.0).
text	string	The picture package custom text. Note: Valid only when <code>content: user</code> . See content .
textColor	RGB color	The color to use for security text.
textPosition	Valid values: centered lowerleft lowerright upperleft upperright	The security text position (default: <code>centered</code>).

Property	Value type	What it is (Continued)
textRotate	Valid values: clockwise 45 clockwise 90 counter clockwise 45 counter clockwise9 0 zero	The orientation to use for security text (default: zero).
typename	string	Read-only. The class name of the referenced object.

pinch

Options that may be supplied when applying a Pinch filter.

The `pinch` class supplies properties for the [filter options](#) class. The `pinch` class is used only to define a record whose properties specify options when applying a Pinch filter. `pinch` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `pinch` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>amount</code>	integer	(-100 - 100)

The following code sample demonstrates how to apply the Pinch filter using the `filter` command of the `pinch` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using pinch filter -
    with options {class:pinch, amount: 40}
```

Pixar save options

Options that can be specified when saving a document in Pixar format. See the [save](#) command for additional details. The `Pixar save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a Pixar file. `Pixar save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `Pixar save options` object.

Properties

Property	Value Type	What it is
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.

The following code creates the `myOptions` variable to hold the `Pixar file save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  set myOptions to {class:Pixar save options, -
    save alpha channels: true}
  save current document in file myFile as Pixar file format with options -
    myOptions appending no extension without copying
end tell
```

PNG save options

Options that can be specified when saving a document in PNG format. See the [save](#) command for additional details. The `PNG save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a PNG file. `PNG save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `PNG save options` object.

Properties

Property	Value Type	What it is
<code>interlaced</code>	boolean	Indication of whether rows should be interlaced (default: <code>false</code>).

The following code creates the `myOptions` variable to hold the `PNG save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  set myOptions to {class:PNG save options}
  save current document in file myFile as PNG format with options ~
    myOptions appending no extension without copying
end tell
```

polar coordinates

Options that may be supplied when applying a Polar Coordinates filter.

The `polar coordinates` class supplies properties for the [filter options](#) class. The `polar coordinates` class is used only to define a record whose properties specify options when applying a Polar Coordinates filter. `polar coordinates` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `polar coordinates` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>kind</code>	Valid Values: polar to rectangular rectangular to polar	The type of polar conversion to use.

The following code sample demonstrates how to apply the Polar Coordinates filter using the `filter` command of the `polar coordinates` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using polar coordinates filter ~
  with options {class:polar coordinates, kind: rectangular to polar}
```

posterize

The `posterize` class supplies options for the [adjustment options](#) class to specify the number of tonal levels for each channel and then map pixels to the closest matching level.

The `posterize` class, which is analogous to the Posterize command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply a Posterize adjustment to an image. The `posterize` class can be used only in conjunction with the `adjust` command. It is not possible to get or create a `posterize` object.

See the [adjust](#) command for additional details.

Properties

Property	Value Type	What it is
<code>levels</code>	integer	(2 - 255)

The following code sample demonstrates how to apply a Posterize adjustment using the `adjust` command of the `posterize` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using mix channels adjustment-
  with options {class:mix channels, output channels:{red:{-42, 62, 5, 116},
  green:{0, 172, 136, 0}, blue: {-66, 0, 100, -112}}}
```

presentation options

Options for the `create PDF presentation` command. See [create PDF presentation](#).

Properties

Property	Value Type	What it is
<code>auto advance</code>	boolean	Indication of whether to auto advance images when viewing the presentation (default: <code>true</code>). Note: Valid only when <code>presentation: true</code> . See presentation .
<code>include filename</code>	boolean	Indication of whether to include the file name for the image (default: <code>false</code>).
<code>interval</code>	integer	The number of seconds before the view is auto advanced (default: 5). Note: Valid only when <code>auto advance: true</code> . See auto advance .
<code>loop</code>	boolean	Indication of whether to loop after the last page of the presentation. Note: Valid only when <code>auto advance: true</code> . See auto advance .
<code>magnification</code>	Valid Values: actual size fit page	The magnification type to use when viewing the image.
<code>pdf file options</code>	object (PDF save options)	Options to use when creating the PDF file.

Property	Value Type	What it is (Continued)
presentation	boolean	<p>Indication of whether the file type is presentation (default: <code>false</code>).</p> <p>Note: <code>false</code> indicates a Multi-Page document.</p>
transition	<p>Valid Values:</p> <ul style="list-style-type: none"> blinds horizontal blinds vertical box in box out dissolve glitter down glitter right glitter right down none random split horizontal in split horizontal out split vertical in split vertical out wipe down wipe left wipe right wipe up 	<p>The transition from one image to the next (default: <code>none</code>).</p> <p>Note: Valid only when <code>auto advance</code>: <code>true</code>. See auto advance.</p>

Valid Commands

[create PDF presentation](#)

radial blur

Options that may be supplied when applying a Radial Blur filter.

The `radial blur` class supplies properties for the [filter options](#) class. The `radial blur` class is used only to define a record whose properties specify options when applying a Radial Blur filter. `radial blur` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `radial blur` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>amount</code>	integer	(0 - 100)
<code>blur method</code>	Valid Values: spin zoom	The type of blur effect to use.
<code>quality</code>	Valid Values: best draft good	

The following code sample demonstrates how to apply the Radial Blur filter using the `filter` command of the `radial blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using radial blur filter ~
  with options {class:radial blur, amount: 35, ~
    blur method:spin, quality:draft}
```

raw format open options

Settings related to opening a raw format document. The `raw format open options` class inherits properties from the [open options](#) class.

This class is used to define a record containing properties that specify options when opening a document in raw format. `raw format open options` can be supplied only in conjunction with the `open` command. It is not possible to get or create a `raw format open options` object.

Properties

Property	Value Type	What it is
<code>bits per channel</code>	integer	The number of bits for each channel (8 or 16).
<code>byte order</code>	Valid Values: IBM PC Mac OS	The order in which bytes will be read. Note: Valid only when <code>bits per channel</code> : 16.
<code>header size</code>	integer	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).
<code>height</code>	integer	The image height (in pixels).
<code>interleave channels</code>	boolean	Indication of whether to store color values sequentially.
<code>number of channels</code>	integer	The number of channels in the image (1 - 56). Note: The value of <code>number of channels</code> cannot exceed the number of channels in the image. When <code>bitsPerChannel</code> : 16, only the following values are valid: 1, 3, or 4. See bits per channel .
<code>retain header</code>	boolean	Indication of whether to retain the header when saving the document. Note: Valid only when header size is 1 or greater.
<code>width</code>	integer	The image width (in pixels).

The following code opens a raw format document.

```
tell application "Adobe Photoshop CS2"
  set myFilePath to alias "Applications:rawFiles:MyFile.pcd"
  open myFilePath as raw with options -
    {class:raw open options, bits per channel:8, -
    header size:144, height:720, width:720-
    number of channels:5, retain header:true-
    interleave channels:true}
end tell
```

raw save options

Options that can be specified when saving a document in raw format. See the [save](#) command for additional details. The `raw save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a raw file. `raw save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `raw save options` object.

Properties

Property	Value Type	What it is
<code>save alpha channels</code>	boolean	Indicates whether alpha channels should be saved.
<code>save spot colors</code>	boolean	Indicates whether the spot colors should be saved.

The following code creates the `myOptions` variable to hold the `raw save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  set myOptions to {class:raw save options, -
    embed color profile: true, save alpha channels: true,-
    save spot colors:false}
  save current document in file myFile as raw format with options -
    myOptions appending no extension without copying
end tell
```

RGB color

An RGB color specification: numeric values for the component colors (red, green, and blue). The `RGB color` class inherits properties from the [color value](#) class.

This class a property of the [color value](#) object and is used to define a record that specifies options when creating a color value.

If the `mode` of a document is RGB and you specify the color value for a stroke, foreground, or background color property using a `CMYK color` object, Adobe Photoshop CS2 translates the CMYK color specification into an RGB color specification. The same thing happens if the document's color mode is CMYK and you specify colors using a `RGB color` object. Since this translation can cause information loss you should specify colors using the color class that matches the document's color mode.

Properties

Property	Value Type	What it is
<code>red</code>	real	The red color value (0.0 - 255.0; default: 255 . 0).
<code>green</code>	real	The green color value (0.0 - 255.0; default: 255 . 0).
<code>blue</code>	real	The blue color value (0.0 - 255.0; default: 255 . 0).

RGB hex color

A hexadecimal specification of an RGB color. The `RGB hex color` class inherits properties from the [color value](#) class.

Properties

Property	Value Type	What it is
<code>hex value</code>	Unicode text	The hex representation of an RGB color.

ripple

Used to apply the Ripple filter. The `ripple` class inherits properties from the [filter options](#) class.

Options that may be supplied when applying a filter to a layer or selection. See the [filter](#) command for additional details.

This class is used to define a record containing properties used to specify options when applying a filter. `ripple` options can only be supplied in conjunction with the `filter` command. It is not possible to get or create a `ripple` object.

Properties

Property	Value Type	What it is
<code>amount</code>	integer	(-999 - 999)
<code>ripple size</code>	Valid Values: large medium small	

The following code sample demonstrates how to apply the Ripple filter using the `filter` command of the `ripple` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using ripple filter -
  with options {class:ripple, amount: 345,-
    ripple size:small}
```

save options

Options used with the save command. The following classes inherit properties from the `save options` class:

[BMP save options](#), [EPS save options](#), [GIF save options](#), [JPEG save options](#), [PDF save options](#), [Photoshop DCS 1.0 save options](#), [Photoshop DCS 2.0 save options](#), [Photoshop save options](#), [PICT file save options](#), [PICT resource save options](#), [Pixar save options](#), [PNG save options](#), [raw save options](#), [SGI RGB save options](#), [Targa save options](#), [TIFF save options](#).

Valid Commands

[save](#)

selection-object

The selection of the document.

Properties

Property	Value Type	What it is
best type	type class	Read-only. The best type for the object's value.
class type	class	Read-only. The object's class.
default type	type class	Read-only. The default type for the object's value.
properties	record	Read-only. All of this object's properties returned in a single record.
container	reference	Read-only. The object's container.

Valid Commands

[clear](#)
[contract](#)
[copy](#)
[copy merged](#)
[cut](#)
[deselect](#)
[expand](#)
[feather](#)
[fill](#)
[grow](#)
[invert](#)
[load](#)
[make work path](#)
[rotate](#)
[rotate boundary](#)
[scale](#)
[scale boundary](#)
[select](#)
[select all](#)
[select border](#)
[similar](#)
[smooth](#)
[store](#)
[stroke](#)
[translate](#)
[translate boundary](#)

selective color

The `selective color` class modifies the amount of a process color in a specified primary color without affecting the other primary colors. The `selective color` class inherits properties from the [adjustment options](#) class.

Properties

Property	Value Type	What it is
<code>selection method</code>	Valid Values: absolute relative	The method to use for interpreting selective color adjustment specifications: <code>absolute</code> = % of the whole; <code>relative</code> = % of the existing color amount.
<code>reds</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>yellows</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>greens</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>cyans</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>blues</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>magentas</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>whites</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>neutrals</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.
<code>blacks</code>	list of reals	Array of 4 values: cyan, magenta, yellow, black.

settings-object

Preferences for the Adobe Photoshop CS2 application.

Properties

Property	Value Type	What it is
<code>best type</code>	type class	Read-only. The best type for the object's value.
<code>class type</code>	class	Read-only. The object's class.
<code>default type</code>	type class	Read-only. The default type for the object's value.
<code>properties</code>	record	Read-only. All of this object's properties returned in a single record.
<code>additional plugin folder</code>	alias	The path to an additional plug-in folder. Valid only when use additional plugin: true. See use additional plugin .
<code>append extension</code>	Valid Values: ask no yes	Save files with extensions on Windows.
<code>ask before saving layered TIFF</code>	boolean	Indication of whether to ask the user to verify layer preservation options when saving a file in TIFF format.
<code>auto update open documents</code>	boolean	Indication of whether to automatically update open documents.
<code>beep when done</code>	boolean	Indication of whether to beep when a process finishes.
<code>cache levels</code>	integer	The number of images to hold in the cache (1 - 8).
<code>color picker</code>	Valid Values: Adobe color picker Apple color picker plug in color picker Windows color picker	
<code>column gutter</code>	real	The width of the column gutters (in points).
<code>column width</code>	real	The width of the columns (in points).
<code>create first snapshot</code>	boolean	Indication of whether to automatically make first snapshot when a new document is created.
<code>display color channels in color</code>	boolean	Indication of whether to display component channels in the Channels palette in color.

Property	Value Type	What it is (Continued)
<code>dynamic color sliders</code>	boolean	Indication of whether to display dynamic sliders in the Channels palette.
<code>edit log items</code>	Valid Values: concise detailed sessiononly	The options for editing history log items. Valid only when <code>use history log: true</code> . See use history log .
<code>export clipboard</code>	boolean	Indication of whether to retain Adobe Photoshop CS2 contents on the clipboard after you exit the application.
<code>font preview size</code>	Valid Values: none preview large preview medium preview small	Indication of whether to show font previews in the type tool font menus.
<code>full size preview</code>	boolean	(Mac only.) Indication of whether to display image previews as full size images or thumbnails.
<code>gamut warning opacity</code>	real	(0 - 100 as percent).
<code>grid size</code>	Valid Values: large medium none small	The size to use for squares in the grid.
<code>grid style</code>	Valid Values: dashed line dotted line solid line	The formatting style for non-printing grid lines.
<code>grid subdivisions</code>	integer	(1 - 100)
<code>guide style</code>	Valid Values: dashed line solid line	The formatting style for non-printing guide lines.
<code>icon preview</code>	boolean	(Mac only.)
<code>image previews</code>	Valid Values: ask no yes	The behavior mode to use when saving files.
<code>interpolation method</code>	Valid Values: bicubic bicubic sharper bicubic smoother bilinear closest neighbor none	The method to use to assign color values to any new pixels created when an image is resampled or resized.
<code>keyboard zoom resizes windows</code>	boolean	Indication of whether to automatically resize the window when zooming in or out using keyboard shortcuts.

Property	Value Type	What it is (Continued)
<code>Mac OS thumbnail</code>	boolean	(Mac only.) Indication of whether to create a thumbnail when saving the image.
<code>maximize compatibility</code>	Valid Values: always ask never	Indication of whether and when to maximize compatibility when opening Adobe Photoshop CS2 (PSD) files.
<code>maximum RAM use</code>	integer	Maximum percentage of available RAM used by Adobe Photoshop CS2 (5 - 100).
<code>nonlinear history</code>	boolean	Indication of whether to allow non-linear history.
<code>number of history states</code>	integer	The number of history states to remember (1 - 100).
<code>other cursors</code>	Valid Values: precise standard	The type of pointer to use.
<code>painting cursors</code>	Valid Values: brush size precise standard	The type of pointer to use.
<code>pixel doubling</code>	boolean	Indication of whether to halve the resolution or (double the size of pixels) to make previews display more quickly.
<code>point size</code>	Valid Values: postscript size traditional size	The point/pica size.
<code>recent file list length</code>	integer	The number of items in the recent file list (0 - 30)
<code>ruler units</code>	Valid Values: cm units inch units mm units percent units pica units pixel units point units	The unit that the scripting system will use when receiving and returning values.
<code>save log items</code>	Valid Values: both log file metadata	Options for saving the history items.
<code>save log items file</code>	alias	The file in which to save the history log.
<code>save palette locations</code>	boolean	Indication of whether to make new palette locations the default location.
<code>show Asian text options</code>	boolean	Indication of whether to display Asian text options in the Paragraph palette.

Property	Value Type	What it is (Continued)
<code>show English font names</code>	boolean	Indication of whether to list Asian font names in English.
<code>show slice numbers</code>	boolean	Indication of whether to display slice numbers in the document window when using the Slice tool.
<code>show tool tips</code>	boolean	Indication of whether to show pop up definitions on mouse over.
<code>smart quotes</code>	boolean	Indication of whether to use curly or straight quote marks.
<code>type units</code>	Valid Values: mm units pixel units point units	The unit type-size that the numeric inputs are assumed to represent.
<code>use additional plugin</code>	boolean	Indication of whether to use an additional folder for compatible plug-ins stored with a different application.
<code>use cache for histograms</code>	boolean	Indication of whether to use the sampled data cache for histograms in the Level dialog (faster but not as accurate).
<code>use diffusion dither</code>	boolean	Indication of whether to use diffusion dithering to minimize distinctive patterning caused by pattern dithering.
<code>use history log</code>	boolean	Turn on and off the history logging
<code>use lowercase extension</code>	boolean	Indication of whether to use lowercase for file extensions.
<code>use Shift key for tool switch</code>	boolean	Indication of whether to enable cycling through a set of hidden tools.
<code>use video alpha</code>	boolean	Indication of whether to enable Adobe Photoshop CS2 to send transparency information to your computer's video board. (Requires hardware support.)
<code>Windows thumbnail</code>	boolean	(Requires hardware support.) Indication of whether to create a thumbnail when saving the image.

SGI RGB save options

Options that can be specified when saving a document in SGI RGB format. See the [save](#) command for additional details. The `SGI RGB save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a SGI RGB file. `SGI RGB save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `SGI RGB save options` object.

Properties

Property	Value Type	What it is
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.
<code>save spot colors</code>	boolean	Indication of whether to save the spot colors.

The following code creates the `myOptions` variable to hold the `SGI RGB save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  set myOptions to {class:SGI RGB save options, -
    save alpha channels: true, save spot colors:true}
  save current document in file myFile as SGI RGB format with options -
    myOptions appending no extension without copying
end tell
```

sharpen

Options that may be supplied when applying a Sharpen filter.

The `sharpen` class supplies properties for the [filter options](#) class. The `sharpen` class is used only to define a record whose properties specify options when using the `filter` command to apply a Sharpen filter. `sharpen` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `sharpen` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the Sharpen filter using the `filter` command of the `sharpen` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using sharpen filter -
  with options {class:sharpen}
```

sharpen edges

Options that may be supplied when applying a Sharpen Edges filter.

The `sharpen edges` class supplies properties for the [filter options](#) class. The `sharpen edges` class is used only to define a record when using the `filter` command to apply a Sharpen Edges filter. `sharpen edges options` can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `sharpen edges` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the Sharpen Edges filter using the `filter` command of the `sharpen edges` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using sharpen edges filter -
  with options {class:sharpen edges}
```

sharpen more

Options that may be supplied when applying a Sharpen More filter.

The `sharpen more` class supplies properties for the [filter options](#) class. The `sharpen more` class is used only to define a record when using the `filter` command to apply a Sharpen More filter. `sharpen more` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `sharpen more` object.

See the [filter](#) command for additional details.

The following code sample demonstrates how to apply the Sharpen More filter using the `filter` command of the `sharpen more` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using sharpen more filter -
  with options {class:sharpen more}
```

shear

Options that may be supplied when applying a Shear filter.

The `shear` class supplies properties for the [filter options](#) class. The `shear` class is used only to define a record whose properties specify options when applying a Shear filter. `shear` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `shear` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
curve	list of points	A list of curve coordinates (x, y) that define the points that describe the shear curve. You must provide x and y coordinates for a minimum of 2 points.
undefined areas	Valid Values: repeat edge pixels wrap around	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.

The following code sample demonstrates how to apply the Shear filter using the `filter` command of the `shear` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using shear filter -
  with options {class:shear, curve:{ {12,4}, {12, 8}, {8, 13}},-
  undefined areas:wrap around}
```

smart blur

Options that may be supplied when applying a Smart Blur filter.

The `smart blur` class supplies properties for the [filter options](#) class. The `smart blur` class is used only to define a record whose properties specify options when applying a Smart Blur filter. `smart blur` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `smart blur` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
radius	real	(0 - 1000)
threshold	real	(0 - 1000)
quality	Valid Values: high low medium	
mode	Valid Values: edge only normal overlay edge	The method to use for smart blurring: edge only, overlay edges: Apply blur only to edges of color transitions; normal: Apply blur to entire image.

The following code sample demonstrates how to apply the Smart Blur filter using the `filter` command of the `smart blur` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using smart blur filter -
  with options {class:smart blur, radius:765, threshold:88,-
    quality:high, mode:normal}
```

spherize

Options that may be supplied when applying a Spherize filter.

The `spherize` class supplies properties for the [filter options](#) class. The `spherize` class is used only to define a record whose properties specify options when applying a Spherize filter. `spherize` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `spherize` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>amount</code>	integer	(-100 - 100)
<code>mode</code>	Valid Values: horizontal normal vertical	The curve (or stretch shape) to use for the distortion.

The following code sample demonstrates how to apply the Spherize filter using the `filter` command of the `spherize` class (inherited from the `filter options` super class).

```
...  
filter current layer of the current document using spherize filter -  
  with options {class:spherize, amount:27, mode:normal}
```

sub path info

Sub path information (returned by the entire `path` property of the [path item](#), [path items](#) class).

Properties

Property	Value Type	What it is
<code>closed</code>	boolean	Indication of whether the path describes an enclosed area.
<code>entire sub path</code>	list of path point info objects	The sub path item's path points.
<code>operation</code>	Valid Values: shape add shape intersect shape subtract shape xor	The sub path's operation on other sub paths.

sub path item, sub path items

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

Elements

Element	Refer to by
<code>path point</code>	numeric index, as a range of elements, before after another element, satisfying a test

Properties

Property	Value Type	What it is
<code>best type</code>	type class	Read-only. The best type for the object's value.
<code>class type</code>	class	Read-only. The object's class.
<code>default type</code>	type class	Read-only. The default type for the object's value.
<code>index</code>	integer	Read-only. The index of this instance of the object.
<code>properties</code>	record	Read-only. All of this object's properties returned in a single record.
<code>closed</code>	boolean	Read-only. Indication of whether the sub path item describes an enclosed area.
<code>container</code>	reference	Read-only. The object's container.
<code>entire sub path</code>	list of path point info objects	Read-only. The sub path item's path points.
<code>operation</code>	Valid Values: shape add shape intersect shape subtract shape xor	The sub path's operation on other sub paths.

Valid Commands

[index](#)

Targa save options

Options that can be specified when saving a document in Targa format. See the [save](#) command for additional details. The `Targa save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a Targa file. `Targa save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `Targa save options` object.

Properties

Property	Value Type	What it is
resolution	Valid Values: sixteen thirty two twenty four	The number of bits per pixel (default: <code>twenty four</code>).
RLE compression	boolean	Indication of whether to use RLE compression (default: <code>true</code>).
save alpha channels	boolean	Indication of whether to save the alpha channels.

The following code creates the `myOptions` variable to hold the `Targa save options`, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
    make new document

    set myOptions to {class:Targa save options, -
        embed color profile: true, save alpha channels: true}
    save current document in file myFile as Targa format with options -
        myOptions appending no extension without copying
end tell
```

text-object

The text item contained in an art layer. A `text-object` is a property of an art layer class whose `kind` property value is `text`. `text-object` is used only to define a record whose properties specify options for a text layer. `text-object` options can be supplied only in conjunction with a text layer; it is not possible to get or create a `text-object` object.

See [art layer](#), [art layers](#) for additional details.

Properties

Property	Value Type	What it is
<code>best type</code>	type class	Read-only. The best type for the object's value.
<code>class type</code>	class	Read-only. The object's class.
<code>default type</code>	type class	Read-only. The default type for the object's value.
<code>properties</code>	record	Read-only. All of this object's properties returned in a single record.
<code>alternate ligatures</code>	boolean	Indication of whether to use alternate ligatures.
<code>antialias method</code>	Valid Values: crisp none sharp smoothing strong	The method of antialiasing to use.
<code>auto kerning</code>	Valid Values: manual metrics optical	The auto kerning option to use.
<code>auto leading</code>	boolean	Indication of whether to use a font's built-in leading information.
<code>auto leading amount</code>	real	The percentage to use for auto (default) leading (0.01 - 5000.00 in points). Valid only when <code>auto leading: true</code> . See auto leading .
<code>baseline shift</code>	real	The baseline offset of the text (unit value).
<code>capitalization</code>	Valid Values: all caps normal small caps	The text case.
<code>container</code>	reference	Read-only. The object's container.
<code>contents</code>	Unicode text	The actual text in the layer.

Property	Value Type	What it is (Continued)
desired glyph scaling	real	<p>The desired amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled).</p> <p>Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification.</p> <p>The following values are also required: minimum glyph scaling and maximum glyph scaling.</p>
desired letter scaling Note: 'letter scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	real	<p>The amount of space between letters (100 - 500; at 0, the width of characters is not scaled).</p> <p>Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification.</p> <p>The following values are also required: minimum letter scaling and maximum letter scaling.</p>
desired word scaling Note: 'word scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	real	<p>The amount (percentage) of space between words (0 - 1000; at 100, no additional space is added between words).</p> <p>Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification.</p> <p>The following values are also required: minimum word scaling and maximum word scaling.</p>
faux bold	boolean	Indication of whether to use faux bold.
faux italic	boolean	Indication of whether to use faux italic.
first line indent	real	The amount (unit value) to indent the first line of paragraphs (-1296 - 1296).
font	Unicode text	The text face of the characters.

Property	Value Type	What it is (Continued)
<code>height</code>	real	The width of the bounding box (unit value) for paragraph text. Note: Valid only when <code>kind: paragraph text</code> . See kind .
<code>horizontal scale</code>	integer	Character scaling (horizontal) in proportion to vertical scale (0 - 1000 in percent). See vertical scale .
<code>horizontal warp distortion</code>	real	The horizontal distortion of the warp (-100 - 100 as percent).
<code>hyphen limit</code>	integer	The maximum number of consecutive lines that can end with a hyphenated word.
<code>hyphenate after first</code>	integer	The number of letters after which hyphenation in word wrap is allowed (1 - 15).
<code>hyphenate before last</code>	integer	The number of letters before which hyphenation in word wrap is allowed (1 - 15).
<code>hyphenate capitalized words</code>	boolean	Indication of whether to allow hyphenation of capitalized words.
<code>hyphenate words longer than</code>	integer	The minimum number of letters a word must have in order for hyphenation in word wrap to be allowed (2 - 25).
<code>hyphenation</code>	boolean	Indication of whether to allow hyphenation in word wrap.
<code>hyphenation zone</code>	real	The distance at the end of a line that will cause a word to break in unjustified type (0 - 720 pica).
<code>justification</code>	Valid Values: center center justified fully justified left left justified right right justified	The paragraph justification.
<code>kind</code>	Valid Values: paragraph text point text	The type of the text wrap to use.

Property	Value Type	What it is (Continued)
language	Valid Values: Brazilian Portuguese Canadian French Danish Dutch English UK English USA Finnish French German Italian Norwegian Nynorsk Norwegian Old German Portuguese Spanish Swedish Swiss German	The language to use.
leading	real	The leading amount (unit value).
left indent	real	The amount (unit value) of space to indent text from the left (-1296 - 1296).
ligatures	boolean	Indication of whether to use ligatures.
maximum glyph scaling	real	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: center justified; justification: fully justified; justification: left justified; OR justification: right justified. See justification . The following values are also required: desired glyph scaling and maximum glyph scaling .
maximum letter scaling Note: 'letter scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).'	real	The maximum amount of space to allow between letters (100 - 500; at 0, the width of characters is not scaled). Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; OR justification: right justified. See justification . The following values are also required: minimum letter scaling and desired letter scaling .

Property	Value Type	What it is (Continued)
<p>maximum word scaling</p> <p>Note: 'word scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).'</p>	real	<p>The maximum amount (percentage) of space to allow between words (0 -1000; at 100, no additional space is added between words).</p> <p>Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification.</p> <p>The following values are also required: minimum word scaling and desired word scaling.</p>
<p>minimum glyph scaling</p>	real	<p>The minimum amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled).</p> <p>Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification.</p> <p>The following values are also required: maximum glyph scaling and desired glyph scaling.</p>
<p>minimum letter scaling</p> <p>Note: 'letter scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).'</p>	real	<p>The minimum amount of space to allow between letters (100 - 500; at 0, the width of characters is not scaled).</p> <p>Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification.</p> <p>The following values are also required: desired letter scaling and maximum letter scaling.</p>

Property	Value Type	What it is (Continued)
minimum word scaling Note: 'word scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS2 application Justification dialog (Select Justification on the Paragraphs palette menu).	real	The minimum amount (percentage) of space to allow between words (0 -1000; at 100, no additional space is added between words). Note: Valid only when justification: center justified; justification: fully justified; justification: left justified; or justification: right justified. See justification . The following values are also required: desired word scaling and maximum word scaling .
no break	boolean	Indication of whether to allow words to break at the end of a line. Tip: When enacted on large amounts of consecutive characters, <code>noBreak: true</code> can prevent word wrap and thus may prevent some text from appearing on the screen.
old style	boolean	Indication of whether to use old style type.
position	list	The position of origin for the text. The list must contain two values (unit value). Tip: Setting the <code>position</code> property is basically equivalent to clicking the text tool at a point in the document to create the point of origin for text.
right indent	real	The amount of space (unit value) to indent text from the right (-1296 - 1296).
Roman hanging punctuation	boolean	Indication of whether to use Roman Hanging Punctuation.
size	real	The font size (in points).
space after	real	The amount of space (unit value) to use after each paragraph (-1296 - 1296).
space before	real	The amount of space (unit value) to use before each paragraph (-1296 - 1296).
strike thru	Valid Values: off strike box strike height	Options for text strikethrough.
stroke color	color value	The text color.

Property	Value Type	What it is (Continued)
<code>text composer</code>	Valid Values: Adobe every line Adobe single line	The composition method to use to evaluate line breaks and optimize the specified hyphenation and justification options. Note: Valid only when <code>kind: paragraph text</code> . See kind .
<code>text direction</code>	Valid Values: horizontal vertical	The text orientation.
<code>tracking</code>	real	The amount of uniform spacing between multiple characters (-1000 - 10000). Tip: 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.
<code>underline</code>	Valid Values: left off right	Options for text underlining.
<code>vertical scale</code>	integer	Character scaling (vertical) in proportion to horizontal scale (0 - 1000 in percent). See horizontal scale .
<code>vertical warp distortion</code>	real	(-100 - 100 as percent).
<code>warp bend</code>	real	(-100 - 100 as percent).
<code>warp direction</code>	Valid Values: horizontal vertical	

Property	Value Type	What it is (Continued)
<code>warp style</code>	Valid Values: arc arc lower arc upper arch bulge fish fish eye flag inflate none rise shell lower shell upper squeeze twist wave	
<code>width</code>	real	The width of the bounding box (unit value) for paragraph text. Note: Valid only when <code>kind: paragraph text</code> . See kind .

Valid Commands

[convert to shape](#)
[create work path](#)
[exists](#)

The following code creates a text item in the art layer `my text`, and then uses the `contents`, `justification`, and `stroke color` properties of the `text-object` class to define the text item.

```

tell application "Adobe Photoshop CS2"
  activate
  set textLayer to make new art layer with properties {name:"my text", kind:text}
  set contents of text object of art layer "my text" to "Hello, World!"
  set justification of text object of art layer "my text" of -
    current document to right
  set stroke color of text object of art layer "my text" to-
    {class:RGB color, red:255, green:0, blue:0}
end tell

```

texture fill

Options that may be supplied when applying a Texture Fill filter.

The `texture fill` class supplies properties for the [filter options](#) class. The `texture fill` class is used only to define a record whose properties specify options when applying a Texture Fill filter. `texture fill` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `texture fill` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>file path</code>	alias	The full path to the texture file. Must be a grayscale Photoshop file.

The following code sample demonstrates how to apply the Texture Fill filter using the `filter` command of the `texture fill` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using texture fill filter ~
    with options {class:texture fill, file path:"Hard Disk:Photoshop
CS2:Textures:TF1"}
```

threshold adjustment

The `threshold adjustment` class supplies options properties for the [adjustment options](#) class to convert 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.

The `threshold adjustment` class, which is analogous to the Threshold command in the Adobe Photoshop CS2 application, is used only to define a record containing properties that specify options when using the `adjust` command to apply a Threshold adjustment to an image. The `threshold adjustment` class can be used only in conjunction with the `adjust` command. It is not possible to get or create an `threshold adjustment` object.

See the [adjust](#) command for additional details.

Properties

Property	Value Type	What it is
<code>level</code>	integer	(1 - 255)

The following code sample demonstrates how to apply a threshold adjustment using the `adjust` command of the `threshold adjustment` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using threshold adjustment-
  with options {class:threshold adjustment, level:36}
```

TIFF save options

Options that can be specified when saving a document in TIFF format. See the [save](#) command for additional details. The `TIFF save options` class inherits properties from the [save options](#) class.

This class is used to define a record containing properties used to specify options when saving a document as a TIFF file. `TIFF save options` can only be supplied in conjunction with the `save` command. It is not possible to get or create a `TIFF save options` object.

Properties

Property	Value Type	What it is
<code>byte order</code>	Valid Values: IBM PC Mac OS	The order in which the document's bytes will be read. (The default is <code>ByteOrder.MACOS</code> when running on Mac OS and <code>ByteOrder.IBM</code> when running on a PC.)
<code>embed color profile</code>	boolean	Indication of whether to embed the color profile in the document.
<code>image compression</code>	Valid Values: JPEG LZW none ZIP	The compression type to use (default: none).
<code>interleaveChannels</code>	boolean	Indication of whether the channels in the image will be interleaved.
<code>JPEG quality</code>	integer	The quality of the produced image (0 - 12), which is inversely proportionate to the amount of JPEG compression. Note: Valid only when <code>image compression: JPEG</code> .
<code>layer compression</code>	Valid Values: RLE ZIP	The method of compression to use when saving layers (as opposed to saving composite data). Note: Valid only when <code>save layers: true</code> . See save layers .
<code>save alpha channels</code>	boolean	Indication of whether to save the alpha channels.
<code>save annotations</code>	boolean	Indication of whether to save the annotations.
<code>save image pyramid</code>	boolean	Indication of whether to preserve multiresolution information (default: false).
<code>save layers</code>	boolean	Indication of whether to save the layers.
<code>save spot colors</code>	boolean	Indication of whether to save the spot colors.
<code>transparency</code>	boolean	Indication of whether to save the transparency as an additional alpha channel when the file is opened in another application.

The following code creates the `myOptions` variable to hold the TIFF save options, then uses the [save](#) command to save the document with the specified options.

```
tell application "Adobe Photoshop CS2"
  make new document

  set myOptions to {class:TIFF save options, -
    embed color profile: true, image compression:JPEG,-
    JPEG quality:4. save layers:true, save spot colors:true}
  save current document in file myFile as TIFF format with options -
    myOptions appending no extension without copying
end tell
```

twirl

Options that may be supplied when applying a Twirl filter.

The `twirl` class supplies properties for the [filter options](#) class. The `twirl` class is used only to define a record whose properties specify options when applying a Twirl filter. `twirl` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `twirl` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>angle</code>	integer	(-999 - 999)

The following code sample demonstrates how to apply the Twirl filter using the `filter` command of the `twirl` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using twirl filter -
  with options {class:twirl, angle:47}
```

unsharp mask

Options that may be supplied when applying an Unsharp Mask filter.

The `unsharp mask` class supplies properties for the [filter options](#) class. The `unsharp mask` class is used only to define a record whose properties specify options when applying an Unsharp Mask filter. `unsharp mask` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create an `unsharp mask` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>amount</code>	real	(1 - 500 as percent)
<code>radius</code>	real	(0.1 - 250.00 in pixels)
<code>threshold</code>	integer	(0 - 255)

The following code sample demonstrates how to apply the Unsharp Mask filter using the `filter` command of the `unsharp mask` class (inherited from the `filter options` super class).

```
...
filter current layer of the current document using unsharp mask filter ~
  with options {class:unsharp mask, amount:437~
    radius:17.8, threshold:255}
```

wave filter

Options that may be supplied when applying a Wave filter.

The `wave filter` class supplies properties for the [filter options](#) class. The `wave` class is used only to define a record whose properties specify options when applying a Wave filter. `wave filter` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `wave` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>number of generators</code>	integer	(1 - 999)
<code>minimum wavelength</code>	integer	(1 - 998)
<code>maximum wavelength</code>	integer	(2 - minimum wavelength + 1)
<code>minimum amplitude</code>	integer	(1 - 998)
<code>maximum amplitude</code>	integer	(2 - minimumAmplitude + 1)
<code>horizontal scale</code>	integer	(1 - 100 as percent)
<code>vertical scale</code>	integer	(1 - 100 as percent)
<code>wave type</code>	Valid Values: sine square triangular	
<code>undefined areas</code>	Valid Values: repeat edge pixels wrap around	The method to use to fill areas left blank by the wave distortion.
<code>random seed</code>	integer	The number to use to control random wave lengths.

The following code sample demonstrates how to apply the Wave filter using the `filter` command of the `wave` class (inherited from the `filter options` super class).

```
filter current layer of the current document using wave filter -
  with options {class:wave filter, number of generators:1 -
    , minimum wavelength:1, maximum wavelength:100, -
    minimum amplitude:5, maximum amplitude:10 -
    , horizontal scale:100, vertical scale:100 -
    , wave type:sine, undefined areas:repeat edge pixels, -
    random seed:0}
```

XMP metadata

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
<code>best type</code>	type class	Read-only. The best type for the object's value.
<code>class type</code>	class	Read-only. The object's class.
<code>default type</code>	type class	Read-only. The default type for the object's value.
<code>properties</code>	record	Read-only. All of this object's properties returned in a single record.
<code>container</code>	reference	Read-only. The object's container.
<code>raw data</code>	Unicode text	The raw XML form of file information.

zigzag

Options that may be supplied when applying a Zigzag filter.

The `zigzag` class supplies properties for the [filter options](#) class. The `zigzag` class is used only to define a record whose properties specify options when applying a Zigzag filter. `zigzag` options can be supplied only in conjunction with the `filter` command. It is not possible to get or create a `zigzag` object.

See the [filter](#) command for additional details.

Properties

Property	Value Type	What it is
<code>amount</code>	integer	(-100 - 100)
<code>ridges</code>	integer	(0 - 20)
<code>style</code>	Valid Values: around center out from center pond ripples	The type of zigzag to use.

The following code sample demonstrates how to apply the Zigzag filter using the `filter` command of the `zigzag` class (inherited from the `filter options` super class).

```
filter current layer of the current document using zigzag filter ↵  
  with options {class:zigzag, amount:-45, ↵  
    ridges:13, style:pond ripples}
```

4

AppleScript Commands

This chapter provides a complete reference for the commands in the Adobe Photoshop CS2 AppleScript dictionary. The commands are presented alphabetically.

The commands supported by each object, with links to the detailed descriptions here, are listed in [‘AppleScript Objects’ on page 10](#).

Overview

This chapter describes the commands in the Adobe Photoshop CS2 AppleScript dictionary, as well as some of the important standard AppleScript commands. The AppleScript dictionary itself shows only that the command returns an object, or that the command takes an object reference as a parameter; it does not show the specific objects that can respond to a particular command. Not all Adobe Photoshop CS2 objects can respond to all commands. This reference details which objects respond to which commands, and what type of object each command returns (if any).

The following information is given for each command.

Column heading	What it means
Parameters	Constants, keywords, and values needed by the command. <ul style="list-style-type: none">• Variable values to be supplied are shown in bold.• Literal terms and constants are shown in plain type.• Items surrounded by brackets [] are optional.
What it is	An explanation of the parameters.
Objects supported	Which objects support the command and/or which objects the command can operate upon. The <code>document</code> object, for example, supports the command <code>close</code> , but not the command <code>quit</code> .
Returns	Many commands return values (text, numbers, lists, and object references). This column shows you what kind of reference you can expect the command to return (if any).

activate

Makes an application active; that is, makes Adobe Photoshop CS2 the frontmost application.

Parameters	What it is	Objects supported	Returns
none		application	nothing

The the first line of the following code sample directs Adobe Photoshop CS2 to carry out the script's commands. The second line uses the `activate` command to make Adobe Photoshop CS2 the frontmost application.

```
tell application "Adobe Photoshop CS2"  
  activate
```

Note: Adobe Photoshop CS2 must be the frontmost application when executing any command that deals with the clipboard. Use `activate` to ensure that Adobe Photoshop CS2 is the active application. See the clipboard commands for examples:

[copy](#)

[cut](#)

[paste](#)

adjust

Applies an adjustment to one or more art layers.

Parameters	What it is	Objects supported	Returns
adjust art layer	The art layer object or objects to be operated upon.	art layer , art layers	
using anything	The options class used to invoke the method. Note: The class must be a class that inherits from adjustment options .	automatic contrast automatic levels brightness and contrast color balance curves desaturate equalize inversion levels adjustment mix channels posterize selective color threshold adjustment	
[with options adjustment options]	The options that can be specified for the command.		

The following code sample demonstrates how to apply an Automatic Contrast adjustment using the `adjust` command of the `automatic contrast` class (inherited from the `adjustment options` super class).

```
...
adjust current layer of the current document using automatic contrast adjustment-
with options {class:automatic contrast}
```

apply

Applies the specified layer comp to the document.

Parameters	What it is	Objects supported	Returns
apply layer comp	The the layer comp object or objects to be operated upon.	layer comp , layer comps	nothing

The following code creates two art layers and a layer comp, then applies the layer comp to the document.

```
tell application "Adobe Photoshop CS2"
  activate

  --create a variable named theDocRef
  --assign the current (active) document to it
  set theDocRef to the current document

  --create two art layers and make a layer comp
  set layer1 to make new art layer in theDocRef with properties ~
    {visible: true}
  set layer2 to make new art layer in theDocRefwith properties ~
    {visible: true}
  set layerComp1 to make new layer comp in theDocRef with properties ~
    {visibility: true, name:firstDocumentState}
  apply LayerComp1
end tell
```

apply layer style

Applies a layer style to the specified art layer.

Parameters	What it is	Objects supported	Returns
apply layer style art layer	The art layer object or objects to which you want to apply the style.	art layer , art layers layer , layers	
using Unicode text	The name of the layer style to apply.		

The following code creates a text layer by creating an art layer and setting its `kind` property to `text layer`, then applies the Drop Shadow layer style to the layer. The layer style is taken from the apply Layer Style list in the Layers palette in the Adobe Photoshop CS2 application.

```
tell application "Adobe Photoshop CS2"
  activate

  --create a variable named theDocRef
  --assign the current (active) document to it
  set theDocRef to the current document

  --create a text layer and apply a style
  set layer1 to make new art layer in theDocRef with properties ~
    {kind:text layer}
  apply layer style layer1 using Drop Shadow
  apply LayerComp1
end tell
```

change mode

Changes the color mode of the document.

Parameters	What it is	Objects supported	Returns
change mode document	The document object or objects to be operated upon.	document , documents	
to bitmap/CMYK/grayscale/ indexed color/Lab/ multichannel/RGB	The mode to change to.		
[with options change mode options]	Options for changing the mode.	Bitmap mode options indexed mode options	

close

Closes a document.

Parameters	What it is	Objects supported	Returns
object reference	The document to close.	document , documents	nothing
[saving yes/no/ask]	Specifies whether to save the document before closing (default: ask).		

► Closing a document

```
-- Close the first document and prompt the user with a "Save as" dialog
tell application "Adobe Photoshop CS2"
    activate
    close document 1 saving ask
end tell
```

clear

Clears the currently selected object.

Parameters	What it is	Objects supported	Returns
clear	Deletes the selected object and does not copy it to the clipboard.	art layer , art layers selection-object	

contract

Contracts the selection.

Parameters	What it is	Objects supported	Returns
<code>contract</code> <code>selection-object</code>	The selection-object object or objects to be operated upon.	selection-object	
by <code>real</code>	The amount to contract the selection (unit value).		

convert

Converts the document from using one color profile to using another.

Parameters	What it is	Objects supported	Returns
convert document	The document object or objects to be operated upon.	document , documents	
to profile Unicode text	The name of the color profile to convert to. Either a string that names a specific color profile or Working RGB, Working CMYK, Working Gray, Lab Color (meaning one of the working color spaces or Lab color).		
intent absolute colorimetric/ perceptual/ relative colorimetric/ saturation intent	The conversion intent.		
[blackpoint compensation boolean]	Indication of whether to use black point compensation.		
[dithering boolean]	Indication of whether to use dither (default: <code>true</code>).		

convert color

Converts an object from one color model to another.

Parameters	What it is	Objects supported	Returns
convert color color value	The color to convert.	color value	color value
to CMYK/grayscale/HSB/ Lab/RGB/RGB Hex	The color model to convert to.		

convert to shape

Converts the text item and its containing layer to a fill layer with the text changed to a clipping path.

Parameters	What it is	Objects supported	Returns
convert to shape text-object	The text-object object or objects to be operated upon.	text-object	

copy

Copies the current selection to the clipboard.

Parameters	What it is	Objects supported	Returns
copy		art layer , art layers selection-object	

copy merged

Copies the current selection, including the data from all visible layers, to the clipboard.

Parameters	What it is	Objects supported	Returns
copy merged		art layer , art layers selection-object	

Note: Commands that manipulate the clipboard (`cut`, `copy`, and `paste`) require that Adobe Photoshop CS2 be the frontmost application during these operations. Use [activate](#) to bring Adobe Photoshop CS2 to the front before executing a `copy` command. No error is returned if there is no selection to copy. If the application is not frontmost, an error is returned.

count

Counts the elements of a specified type contained in a specified object.

Parameters	What it is	Objects supported	Returns
count object reference	The object whose elements are to be counted.	art layer , art layers channel , channels document , documents	integer
[each type class]	The class of the objects to count.	font , fonts history state , history states layer , layers layer comp , layer comps layer set , layer sets notifier , notifiers path item , path items path point , path points sub path item , sub path items	

Notes

With the optional `each` term, use the singular form for the object type to be counted; for example, `art layer` rather than `art layers`. If you don't use `each`, you can use the singular or plural form.

► Counting filled path items in a document

```
-- This script shows the user how many paths
-- are filled out of the total number in document 1
tell application "Adobe Photoshop CS2"
    set pathCount to count every path item of document 1
    set numberFilled to -
        count (path items of document 1 whose filled is true)
    display dialog numberFilled & " of " & pathCount -
        & " paths are filled in this document." as string
end tell
```

create PDF presentation

Creates a PDF presentation file.

Parameters	What it is	Objects supported	Returns
from files list of alias	A list of the input files to include in the presentation.		PDF presentation file
to file file specification	The location of the output file.		
[with options presentation options]	Options that can be specified for the PDF presentation.	presentation options	

create photo gallery

Creates a Web photo gallery.

Parameters	What it is	Objects supported	Returns
create photo gallery		gallery options	Reference to Web photo gallery
from folder alias	The folder of files to include in the photo gallery.		
to folder alias	The location for the output files.		
[with options gallery options]	Options that can be specified for the web photo gallery.	gallery banner options gallery color options gallery images options gallery security options gallery thumbnail options	

create selection

Makes a selection object from the specified path.

Parameters	What it is	Objects supported	Returns
create selection path item	The path item object or objects to be operated upon.	path item, path items	selection-object
[feather amount real]	The feather amount in pixels.		
[antialiasing boolean]	Indication of whether to use anti aliasing for the selection.		
[operation diminished/extended/intersected/replaced]	The operation mode for the selection.		

create work path

Creates a work path based on the text object.

Parameters	What it is	Objects supported	Returns
<code>create work path</code> <code>text-object</code>	The text-object object or objects to be operated upon.	text-object	

crop

Crops the document.

Parameters	What it is	Objects supported	Returns
crop document	The document object or objects to be operated upon.	document , documents	
bounds list	The area to crop (unit value).		
[angle real]	The angle of the cropping bounds (default: 0 . 0).		
[width real]	The width of the resulting document (unit value).		
[height real]	The height of resulting document (unit value).		

cut

Cuts the current selection to the clipboard.

Parameters	What it is	Objects supported	Returns
cut	Deletes the selected object and copies it to the clipboard.	art layer , art layers , selection-object	

Note: Commands that manipulate the clipboard (`cut`, `copy`, and `paste`) require that Adobe Photoshop CS2 be the frontmost application during these operations. Use [activate](#) to bring Adobe Photoshop CS2 to the front before executing a `copy` command. No error is returned if there is no selection to copy. If the application is not frontmost, an error is returned.

delete

Removes an element from an object.

Parameters	What it is	Objects supported	Returns
<code>object reference</code> or list (of <code>object references</code>)	Contained object or objects to delete.	art layer , art layers channel , channels layer , layers layer comp , layer comps layer set , layer sets notifier , notifiers path item , path items	nothing

► Deleting a layer

```
-- This script deletes the second layer in the document
tell application "Adobe Photoshop CS2"
    if (count layers of document 1) > 1 then
        delete layer 2 of document 1
    end if
end tell
```

deselect

Deselects the selected object.

Parameters	What it is	Objects supported	Returns
deselect object	The object or objects to be operated upon.	document , documents path item , path items selection-object	

do action

Plays an action from the Actions Palette.

Parameters	What it is	Objects supported	Returns
<code>do action Unicode text</code>	The name of the action to play. Note: The Action name is case sensitive. Be sure to enter the Action name exactly as it appears in the Actions palette.		
<code>from Unicode text</code>	The name of the Action Set containing the Action to be played. Note: The Action Set name is case sensitive. Be sure to enter the Action Set name exactly as it appears in the Actions palette.		

do javascript

Executes a JavaScript.

Parameters	What it is	Objects Supported	Returns
do javascript anything	The JavaScript code or file (.js or .jsx) to execute.		Unicode text
[with arguments list of anything]	The arguments to pass to the JavaScript routine.		
[show debugger before running/ never/ on runtime error]	Indication of when, if at all, to show the debugger (default: never).		

duplicate

Duplicates one or more objects.

Parameters	What it is	Objects supported	Returns
[to]	location reference	art layer , art layers channel , channels document , documents	reference to the duplicated object
[with properties]	record	layer , layers layer set , layer sets path item , path items	

equal colors

Determines whether the specified colors are visually equal.

Parameters	What it is	Objects supported	Returns
<code>equal colors color value</code>	The cPsBaseColor object to be operated upon.	color value	boolean
<code>with color value</code>	The color to which to compare the original color.		

exists

Determines whether an object exists.

Parameters	What it is	Objects supported	Returns
<code>object reference</code>	The object to test for existence.	Any object except <code>application</code>	<code>boolean</code>

► Check if a document exists

```
-- Check if a document exists and beep twice if one does
tell application "Adobe Photoshop CS2"
    if exists document 1 then beep 2
end if
end tell
```

expand

Expands the current selection.

Parameters	What it is	Objects supported	Returns
expand selection-object	The selection-object object or objects to be expanded.	selection-object	
by real	The amount by which to expand the selection (unit value).		

export

Exports the document.

Parameters	What it is	Objects supported	Returns
export document	The document object or objects to be operated upon.	document , documents	
in file specification	The file to export to.		
[as Illustrator paths]	The type of export (default: Illustrator paths).		
[with options Illustrator paths export options]	Options that can be specified for the export type.		

feather

Feathers the edges of the current selection.

Parameters	What it is	Objects supported	Returns
<code>feather</code> <code>selection-object</code>	The object or objects to be operated upon.	selection-object	
by <code>real</code>	The amount of the edge to feather (unit value).		

fill

Fills the current selection.

Parameters	What it is	Objects supported	Returns
fill selection-object	The selection-object object or objects to be operated upon.	selection-object	
with contents anything	A color object or history state object.	CMYK color color value gray color history state, history states HSB color RGB color RGB hex color	
[blend mode behind mode/ clear mode/ color blend/ color burn/ color dodge/darken/ difference/dissolve/ exclusion/hard light/ hue blend/lighten/ linear burn/ linear dodge/ linear light/ luminosity/ multiply/normal/ overlay/ pin light/ saturation blend/ screen/soft light/ vivid light]	The color blend mode to use (default: normal).		
[opacity integer]	The fill opacity (default: 100 as percent).		
[preserving transparency boolean]	Indication of whether to preserve the transparencies.		

fill path

Fills the path with the specified information.

Parameters	What it is	Objects supported	Returns
fill path path item	The path item object or objects to be operated upon.	path item, path items	
[with contents anything]	The color of the fill for this path.	CMYK color color value gray color history state, history states HSB color RGB color RGB hex color	

Parameters	What it is	Objects supported	Returns
[blend mode behind mode/clear mode/ color blend/color burn/ color dodge/darken/ difference/dissolve/ exclusion/hard light/ hue blend/lighten/ linear burn/linear dodge/ linear light/luminosity/ multiply/normal/overlay/ pin light/ saturation blend/ screen/soft light/ vivid light]	The blending mode to use for the path's fill.		
[opacity real]	The opacity of the path's fill.		
[antialiasing boolean]	Indication of whether to use anti aliasing for the selection.		
[preserving transparency boolean]	Indication of whether to preserve the transparency of the fill for this path.		
[feather amount real]	The feather amount in pixels.		
[antialiasing boolean]	Indication of whether to use anti aliasing for the selection.		

filter

Applies a filter to one or more art layers.

Parameters	What it is	Objects supported	Returns
<code>filter art layer</code>	The art layer object or objects to be operated upon.	art layer , art layers layer , layers	
using anything	The options class used to invoke the method. Note: The class must be a class that inherits from filter options .		
[with options filter options]	The options that can be specified for the method.		

flatten

Flattens all layers in the document.

Parameters	What it is	Objects supported	Returns
<code>flatten document</code>	The document object or objects to be operated upon.	document , documents	

flip canvas

Flips the canvas horizontally or vertically.

Parameters	What it is	Objects supported	Returns
flip canvas document	The document object or objects to be operated upon.	document , documents	
direction horizontal/vertical	The direction in which to flip the canvas.		

grow

Grows the current selection to include all adjacent pixels falling within the specified tolerance range.

Parameters	What it is	Objects supported	Returns
grow selection-object	The selection-object object or objects to be operated upon.	selection-object	
tolerance integer	(0 - 255)		
antialiasing boolean			

import annotations

Imports annotations into the document.

Parameters	What it is	Objects supported	Returns
<code>import annotations document</code>	The document object or objects to be operated upon.	document , documents	
<code>from alias</code>	The document from which to import the annotations.		

index

Gets an element from the specified element.

Parameters	What it is	Objects supported	Returns
item key	The object or objects to be operated upon.	art layer , art layers channel , channels document , documents font , fonts history state , history states layer , layers layer comp , layer comps layer set , layer sets notifier , notifiers path item , path items path point , path points sub path item , sub path items	object reference

invert

Inverts the selection.

Parameters	What it is	Objects supported	Returns
<code>invert</code> <code>selection-object</code>	The selection-object object or objects to be operated upon.	art layer , art layers layer , layers selection-object	

link

Links the layer with another layer.

Parameters	What it is	Objects supported	Returns
link layer	The layer object or objects to be operated upon.	art layer , art layers , layer , layers , layer set , layer sets	
with layer	The layer to link to.		

load

Loads a selection from a channel.

Parameters	What it is	Objects supported	Returns
load selection-object	The selection-object object or objects to be operated upon.	selection-object	
from channel	The channel from which to load the selection.	channel, channels	
[combination type diminished/extended/ intersected/replaced]	The method to use to combine the channel contents with the existing selection (default: replaced).		
[inverting boolean]	Indication of whether to select the inverse of the channel contents (default: <code>false</code>).		

make

Creates a new object and returns a reference to newly created object.

Parameters	What it is	Objects supported	Returns
<code>new type class</code>	The class of object to create. Note: The term <code>new</code> is optional.	art layer , art layers channel , channels document , documents layer , layers	object reference
<code>at location reference</code>	Location at which to insert new object.	layer comp , layer comps layer set , layer sets	
[with properties record]	Any property of the object you wish to set at creation.	notifier , notifiers path item , path items	
[with data anything]	Any data needed for creation that is not a property.		
[event class four-character ID or unique string]	Defines the class ID of the event as a four character ID or a unique string. See Appendix A: Event ID Codes . Tip: Remember to omit the single quotes when including a four-character ID in your code. Note: An <code>eventClass</code> 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.	notifier , notifiers	

► Creating layer objects

```
-- Make 2 layers, one at the top and one at the bottom
-- demonstrating the power of location references like beginning and end

tell application "Adobe Photoshop CS2"
    set topLayer to make new layer -
        at beginning of document 1 with properties {name"Top Layer"}
    set bottomLayer to make new layer -
        at end of document 1 with properties {name"Bottom Layer"}
end tell
```

make clipping path

Makes the current path item the clipping path for this document.

Parameters	What it is	Objects supported	Returns
make clipping path path item	The path item object or objects to be operated upon.	path item , path items	
[flatness real]			

make work path

Makes the current selection item the work path for the document.

Parameters	What it is	Objects supported	Returns
make work path selection-object	The selection-object object or objects to be operated upon.	selection-object	
[tolerance real]	The tolerance in pixels.		

merge

Merges an art layer into or layer set into the layer below, or merges a spot channel into the component channels. Merging a layer or layer set returns a reference to the resulting layer. Merging a channel does not return any value.

Parameters	What it is	Objects supported	Returns
merge art layer/channel/layer/ layer set	The object or objects to be operated upon.	art layer , art layers , channel , channels , layer , layers , layer set , layer sets	Merging a layer or layer set returns a reference to the resulting layer. Merging a channel does not return any value.

merge visible layers

Flattens all visible layers in the document.

Parameters	What it is	Objects supported	Returns
merge visible layers document	The document object or objects to be operated upon.	document , documents	

move

Moves one or more objects to a new location; returns references to the moved object or objects at the new location.

Parameters	What it is	Objects supported	Returns
<code>object reference</code> or list (of <code>object references</code>)	Object or objects to move.	art layer , art layers layer , layers layer set , layer sets	object reference or list (of object references)
<code>to location reference</code>	New location of the object or objects.		

Notes

Objects cannot be moved between documents.

► Moving objects to a layer

```
-- This script moves all objects in a document to the first layer
tell application "Adobe Photoshop CS2"
    set allPageItems to every page item of document 1
    move allPageItems to beginning of layer 1 of document 1
end tell
```

► Moving layers

```
-- This script moves the bottommost layer to after the first layer
tell application "Adobe Photoshop CS2"
    move last layer of document 1 to after first layer
end tell
```

open

Opens one or more specified document file(s).

Parameters	What it is	Objects supported	Returns
<code>file specification</code>	The file to be opened.	N/A	nothing
as Acrobat TouchUp Image/ Alias PIX/BMP/camera raw/ CompuServe GIF/ Electric Image/EPS/ EPS PICT preview/ EPS TIFF preview/Filmstrip/JPEG/ PCX/PDF/Photo CD/ Photoshop DCS 1.0/ Photoshop DCS 2.0/ Photoshop EPS/ Photoshop format/ Photoshop PDF/PICT file/ PICT resource/Pixar/PNG/ Portable Bitmap/raw/ Scitex CT/SGI RGB/ SoftImage/Targa/ TIFF/Wavefront RLA/ Wireless Bitmap			
[showing dialogs always/error dialogs/never]	The type of dialogs, if any, to allow when the script is run.		
with options open options	Options for opening a particular type of file.	camera raw open options EPS open options PDF open options Photo CD open options raw format open options	

► Opening a PDF file

This script sample opens a PDF document.

```

...
tell application "Adobe Photoshop CS2"
    set myFilePath to alias "Applications:PDFFiles:MyFile.pdf"
    open myFilePath as PDF with options ~
        {class:PDF open options, height:pixels 100, ~
            width:pixels 200, mode:RGB, resolution:72, ~
            use antialias:true, page:3, ~
            constrain proportions:false}
end tell
    
```

paste

Paste the clipboard contents into the current document.

Parameters	What it is	Objects supported	Returns
[clipping to selection boolean]	Indication of whether to paste the clipboard contents into the current selection (default: <code>false</code>).	document , documents	

Note: Commands that manipulate the clipboard (`cut`, `copy`, and `paste`) require that Adobe Photoshop CS2 be the frontmost application during these operations. Use [activate](#) to bring Adobe Photoshop CS2 to the front before executing a `copy` command. No error is returned if there is no selection to copy. If the application is not frontmost, an error is returned.

print

Prints one or more documents or files.

Parameters	What it is	Objects supported	Returns
anything	Document or list of documents, or file or list of files to be printed.	document , documents	nothing
[postscript encoding ASCII/binary/JPEG]	The encoding type (default: binary)		
[source space document space/proof space]	The color space for source (default: document space).		
[postscript encoding ASCII/binary/JPEG]	The encoding type (default: binary).		
[print space Unicode text]	The color space for the printer. Can be 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: <i>nothing</i>).		
[intent absolute colorimetric/perceptual/relative colorimetric/saturation intent]	The color conversion intent (default: relative colorimetric). Note: Valid only when the print space is different than the source space.		
[blackpoint compensation boolean]	Indication of whether the color conversion should use black point compensation (default: true). Note: Valid only when the print space is different than the source space.		

► Print a document

```
-- Print the current document without displaying a dialog
```

```
tell application "Adobe Photoshop CS2"
    print document 1 without dialog
end tell
```

purge

Purges one or more caches.

Parameters	What it is	Objects supported	Returns
<code>purge</code> all caches/ clipboard cache/ history caches/ undo caches	The caches to purge.	application	

quit

Forces Adobe Photoshop CS2 to quit.

Parameters	What it is	Objects supported	Returns
none		application	nothing

Notes

If there is Adobe Photoshop CS2 data on the clipboard, Adobe Photoshop CS2 displays a dialog asking if you want to save the clipboard for other applications. To prevent this dialog from being displayed, send the following command to the frontmost application

```
set the clipboard to {}
```

► Quitting Adobe Photoshop CS2

```
tell application "Adobe Photoshop CS2"
  activate
  set the clipboard to {}
  close every document saving no
  quit
end tell
```

rasterize

Rasterizes the contents of an art layer.

Parameters	What it is	Objects supported	Returns
<code>rasterize art layer</code>	The art layer object or objects to be operated upon.	art layer , art layers layer , layers	
<code>affecting all linked layers/ entire layer/ fill content/ layer clipping path/ shape/text contents</code>	The layers or layer contents or components to rasterize.		

recapture

Recaptures the current layer state(s) for this layer comp.

Parameters	What it is	Objects supported	Returns
recapture layer comp	The layer comp object or objects to be operated upon.	layer comp , layer comps	

reset from comp

Resets the layer comp state to the document state.

Parameters	What it is	Objects supported	Returns
<code>reset from comp layer comp</code>	The layer comp object or objects to be operated upon.	layer comp , layer comps	

resize canvas

Changes the size of the canvas.

Parameters	What it is	Objects supported	Returns
<code>resize canvas document</code>	The document object or objects to be operated upon.	document , documents	
[width real]	The desired width of the canvas (unit value).		
[height real]	The desired height of the canvas (unit value).		
[anchor position bottom center/ bottom left/ bottom right/ middle center/ middle left/ middle right/ top center/top left/ top right]	The anchor to resize around (default: middle center).		

resize image

Changes the size of the image.

Parameters	What it is	Objects supported	Returns
<code>resize image document</code>	The document object or objects to be operated upon.	document , documents	
[width real]	The desired width of the canvas (unit value).		
[height real]	The desired height of the canvas (unit value).		
[resolution real]	The resolution (in pixels per inch).		
[resample method bicubic/ bicubic sharper/ bicubic smoother/ bilinear/ closest neighbor/ none]	The method to use for resampling pixels (default: bicubic).		

reveal all

Expands the document to show clipped sections.

Parameters	What it is	Objects supported	Returns
reveal all document	The document object or objects to be operated upon.	document , documents	

rotate

Rotates the object.

Parameters	What it is	Objects supported	Returns
rotate layer	The object or objects to be operated upon.	art layer , art layers layer , layers layer set , layer sets selection-object	
angle real	The number of degrees to rotate. A positive angle will rotate the canvas clockwise; a negative value will rotate the canvas counter-clockwise.		
[anchor position bottom center/ bottom left/ bottom right/ middle center/ middle left/ middle right/ top center/ top left/top right]	The anchor point around which to rotate the layer (default: middle center).		

rotate boundary

Rotates the boundary of the selection-object.

Parameters	What it is	Objects supported	Returns
rotate boundary selection-object	The selection-object object or objects to be operated upon.	selection-object	
angle real	The number of degrees to rotate. A positive angle will rotate the canvas clockwise; a negative value will rotate the canvas counter-clockwise.		
[anchor position bottom center/ bottom left/ bottom right/ middle center/ middle left/ middle right/ top center/ top left/top right]	The anchor point around which to rotate the layer (default: middle center).		

rotate canvas

Rotates the canvas.

Parameters	What it is	Objects supported	Returns
rotate canvas document	The document object or objects to be operated upon.	document , documents	
angle real	The number of degrees to rotate. A positive angle will rotate the canvas clockwise; a negative value will rotate the canvas counter-clockwise.		

save

Saves a Adobe Photoshop CS2 document. Returns a reference to the saved document.

Parameters	What it is	Objects supported	Returns
<code>document</code>	The document to save.	document , documents	object reference
[in file <code>specification</code>]	The file to save to, specified as a string containing the full file path or an alias. If not specified, the document is saved to its existing file.		
[as Alias PIX/BMP/ CompuServe GIF/ Electric Image/JPEG/PCX/ Photoshop DCS 1.0/ Photoshop DCS 2.0/ Photoshop EPS/ Photoshop format/ Photoshop PDF/ PICT file/ PICT resource/Pixar/PNG/ Portable Bitmap/raw/ Scitex CT/SGI RGB/ SoftImage/Targa/TIFF/ Wavefront RLA/ Wireless Bitmap]	The file type to which to save.		
[copying <code>boolean</code>]	save as copy leaving original open. Unsaved files cannot be saved as copy.		
[appending lowercase extension/ no extension/ uppercase extension]	append an extension to the document name		
[with options save options]	The save options for the specified file type.	BMP save options EPS save options GIF save options JPEG save options PDF save options Photoshop DCS 1.0 save options Photoshop DCS 2.0 save options Photoshop save options PICT file save options PICT resource save options Pixar save options PNG save options raw save options SGI RGB save options Targa save options TIFF save options	

This example shows to batch process folders of Adobe Photoshop CS2 documents, saving each as a PDF file with specific settings.

```
-- Process all files in folders dropped on this script
-- (when saved as an applet)
-- Save each Adobe Photoshop CS2 file as a PDF file.
on run
    tell me to open {choose folder}
end run

on open droppedItems
    set destFolder to choose folder with prompt "Destination folder?"
    repeat with anItem in droppedItems
tell application "Finder"
    -- Make sure each item processed by this script is a folder
    if class of item anItem is not folder then
    -- Not a folder, notify the user of the error
        display dialog "Please drop only folders on this script"
    else
    -- A folder, get the Adobe Photoshop CS2 files and process them
        set fileList to (every file of anItem whose creator type is ~
            "ART5") as alias list
        end if
    end tell
    SaveFilesAsPDF(fileList, destFolder)
end repeat
end open

-- fileList is a list of aliases to Photoshop files
-- destFolder is an alias to a folder where the PDF files are to be saved
on SaveFilesAsPDF(fileList, destFolder)
    set destPath to destFolder as string
    repeat with aFile in fileList
tell application "Finder" to set fileName to name of aFile
        set newPath to destPath & fileName & ".pdf"
tell application "Adobe Photoshop CS2"
            open aFile
            save current document in file newPath as pdf ~
                with options {classPDF save options, ~
                    compatibilityAcrobat 5, preserve editabilitytrue}
            close current document saving no
        end tell
    end repeat
end SaveFilesAsPDF
```

scale

Scales the object to the specified size.

Parameters	What it is	Objects supported	Returns
<code>scale layer</code>	The object or objects to be operated upon.	art layer , art layers layer , layers layer set , layer sets selection-object	
[horizontal scale real]	The amount to scale the layer horizontally (default: 100.0 as percent).		
[vertical scale real]	The amount to scale the layer (default: 100.0 as percent).		
[anchor position bottom center/ bottom left/ bottom right/ middle center/ middle left/ middle right/ top center/top left/ top right]	The anchor point around which to rotate the layer (default: middle center).		

scale boundary

Scales the boundary of selection.

Parameters	What it is	Objects supported	Returns
<code>scale boundary selection-object</code>	The selection-object object or objects to be operated upon.	selection-object	
[horizontal scale real]	The amount to scale the layer horizontally (default: 100.0 as percent).		
[vertical scale real]	The amount to scale the layer (default: 100.0 as percent).		
[anchor position bottom center/ bottom left/ bottom right/ middle center/ middle left/ middle right/ top center/top left/ top right]	The anchor point around which to rotate the layer (default: middle center).		

select

Changes the selection.

Parameters	What it is	Objects supported	Returns
<code>select object</code>	The object or objects to be operated upon.	path item , path items selection-object	
<code>region list of anything</code>	A list of corners of the selection (in pixels).		
[combination type diminished/extended/intersected/replaced]	The method to use to combine the new selection with the existing selection (default: replaced).		
[feather amount real]	The feather amount (default: 0.0).		
[antialiasing boolean]	Indication of whether to use antialiasing (default: <code>true</code>).		

select all

Selects the entire image.

Parameters	What it is	Objects supported	Returns
<code>select all document</code>	The document object or objects to be operated upon.	document , documents selection-object	

select border

Selects the border of the selection.

Parameters	What it is	Objects supported	Returns
<code>select border selection-object</code>	The selection-object object or objects to be operated upon.	selection-object	
<code>width real</code>	The width of the selection (unit value).		

similar

Grows the selection to include pixels throughout the image falling within the tolerance range.

Parameters	What it is	Objects supported	Returns
<code>similar</code> <code>selection-object</code>	The selection-object object or objects to be operated upon.	selection-object	
tolerance <code>integer</code>			
antialiasing <code>boolean</code>			

smooth

Cleans up stray pixels left inside or outside a color-based selection within the radius specified in pixels.

Parameters	What it is	Objects supported	Returns
<code>smooth</code> <code>selection-object</code>	The selection-object object or objects to be operated upon.	selection-object	
<code>radius</code> integer	The sample radius in pixels (0 - 100).		

split channels

Splits the document's channels.

Parameters	What it is	Objects supported	Returns
<code>split channels</code> <code>document</code>	The document object or objects to be operated upon.	document , documents	list of document objects

store

Saves the selection as a channel.

Parameters	What it is	Objects supported	Returns
store selection-object	The selection-object object or objects to be operated upon.	selection-object	
into channel	The channel to save the selection to.		
[combination type diminished/extended/ intersected/replaced]	The method to use to add the selection to the existing contents of the channel (default: replaced).		

stroke

Strokes the selection.

Parameters	What it is	Objects supported	Returns
stroke selection-object	The selection-object object or objects to be operated upon.	selection-object	
using color anything	The stroke color.		
width integer	The stroke width (in pixels).		
[location center/inside/outside]	The stroke placement (default: center).		
[blend mode behind mode/clear mode/ color blend/color burn/ color dodge/darken/ difference/dissolve/ exclusion/hard light/ hue blend/lighten/ linear burn/ linear dodge/ linear light/luminosity/ multiply/normal/overlay/ pin light/ saturation blend/ screen/soft light/ vivid light]	The color blending mode to use (default: normal).		
[opacity integer]	The opacity of the stroke (default: 100 as percent).		
[preserving transparency boolean]			

stroke path

Strokes the path with the specified information.

Parameters	What it is	Objects supported	Returns
stroke path path item	The path item object or objects to be operated upon.	path item , path items	
[tool art history brush tool/ background eraser tool/ blur tool/brush tool/ burn tool/ clone stamp tool/ color replacement tool/ dodge tool/eraser tool/ healing brush tool/ history brush tool/ pattern stamp tool/ pencil tool/sharpen tool/ smudge tool/sponge tool]	The tool to use when stroking the path.		
[simulate pressure boolean]	Indication of whether to simulate the pressure when using the tool.		

translate

Moves a layer or selection the specified amount (in pixels) relative to its current position.

Parameters	What it is	Objects supported	Returns
translate layer	The object or objects to be operated upon.	art layer , art layers layer , layers layer set , layer sets selection-object	
[delta x real]	The horizontal transformation (unit value).		
[delta y real]	The vertical transformation (unit value).		

translate boundary

Moves the selection relative to its current position.

Parameters	What it is	Objects supported	Returns
<code>translate boundary</code> <code>selection-object</code>	The selection-object object or objects to be operated upon.	selection-object	
[delta x real]	The horizontal transformation (unit value).		
[delta y real]	The vertical transformation (unit value).		

trap

Applies trapping to a CMYK document.

Parameters	What it is	Objects supported	Returns
trap document	The document object or objects to be operated upon.	document , documents	
width integer	The trap width in pixels.		

trim

Trims the transparent area around the image on the specified sides of the canvas.

Parameters	What it is	Objects supported	Returns
trim document	The document object or objects to be operated upon.	document , documents	
[basing trim on bottom right pixel/ top left pixel/ transparent pixels]	The color of, or type of, pixels to base trim on (default: top left pixel).		
[top trim boolean]	Indication of whether to trim away the top of document (default: true).		
[left trim boolean]	Indication of whether to trim away the left of document (default: true).		
[bottom trim boolean]	Indication of whether to trim away the bottom of document (default: true).		
[right trim boolean]	Indication of whether to trim away the right of document (default: true).		

unlink

Unlinks the layer.

Parameters	What it is	Objects supported	Returns
unlink layer	The layer object or objects to be operated upon.	art layer , art layers layer , layers layer set , layer sets	

web safe color

Finds the closest web safe color for a color.

Parameters	What it is	Objects supported	Returns
for <code>color value</code>	The color to be changed.	color value CMYK color gray color HSB color Lab color no color RGB color RGB hex color	RGB color

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	'DrfC'

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	'GlowE'
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	'Ttxt'
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

- A**
- Action Manager
 - defined 9
 - actions
 - palette 9
 - playing 188
 - Add Noise filter 11
 - adjust command
 - options for 12
 - using 172
 - adjustment options 12
 - Adobe Photoshop CS 2, changes 8
 - anchor points 106, 107
 - annotations 163, 202
 - application
 - activating 14, 171
 - bringing to front 13
 - colors 13
 - preference settings 14
 - properties 13
 - quitting 215
 - sample code 14
 - version 14
 - art layers
 - See layers
 - auto kerning 153
 - auto leading 153
- B**
- background color
 - application 13
 - as fill 48
 - galleries 61
 - RGB documents 34
 - background layer 16, 47
 - baseline shift 153
 - Batch command 20
 - beep 139, 192
 - bitmap documents
 - opening 211
 - saving 25
 - specifying options for 22
 - bitmap images See bitmap documents
 - black and white images 162
 - Blur filter 23
 - blur filters
 - Blur 23
 - Blur More 24
 - Gaussian Blur 67
 - Glass 70
 - Lens Blur 89
 - Motion Blur 97
 - Radial Blur 130
 - Smart Blur 148
 - Blur More filter 24
 - BMP documents
 - saving 136
 - BMP documents See bitmap documents
 - brightness
 - equalizing 55
 - specifying options 27
- C**
- caches
 - data 142
 - levels 139
 - purging 214
 - camera raw documents
 - opening 28, 103, 211
 - canvas
 - flipping 200
 - resizing 219
 - trimming 235
 - captions
 - contact sheets 37
 - document 78
 - gallery images 62
 - gallery thumbnails 66
 - picture packages 121
 - change mode command 175
 - indexed mode 76
 - options 30
 - changes in Photoshop CS2 8
 - channels
 - activating 48
 - colors 31
 - creating 207
 - current 48
 - displaying in color 139
 - loading from 206
 - merging 209
 - mixing 96
 - specifying type 31
 - splitting 229
 - clipboard commands 171
 - clipping paths
 - defining 104
 - from path 207
 - from text 179
 - close command 176
 - Clouds filter 33
 - CMYK color 34
 - color picker 139
 - colors
 - adjusting 172
 - balancing 35
 - changing mode 175
 - channels 31
 - comparing 191
 - converting 179
 - correcting 35
 - inverting 80
 - modifying 138
 - none 97
 - process colors 138

- text 158
 - web save 237
- column gutter 139
- commands
 - clipboard 171
 - See individual commands
- CompuServe GIF documents
 - opening 211
- Compuserve GIF documents
 - saving 68
- contact sheets
 - columns 37
 - dimensions 38
 - rows 37
- contrast
 - adjusting 172
 - automatic 18
 - specifying options 27
- copying 180
- copyright
 - document 78
 - gallery images 62
- counting 181
- credits
 - documents 78
 - gallery images 62
 - gallery thumbnails 66
 - picture packages 121
- cropping 183
- cursors 141
- curved segments (paths) 107, 150, 151
- curves
 - adjusting 39, 172
 - in paths 107
- Custom filter 40

D

- DCS 1.0 documents
 - opening 114, 211
 - saving 136
- DCS 2.0 documents
 - opening 211
 - saving 116, 136
- De-Interlace filter 41
- delete command 186
- deleting, objects 186
- desaturate command 42
- Despeckle filter 43
- dialogs
 - display modes 13
 - suppressing 21
 - types 13
- Difference Clouds filter 44
- Diffuse Glow filter 45
- Displace filter 46
- distort filters
 - Diffuse Glow 45
 - Displace 46
 - Ocean Ripple 101

- Pinch 123
- Polar Coordinates 126
- Ripple 135
- Shear 147
- Spherize 149
- Twirl 165
- Wave 167
- Zigzag 169
- documents
 - closing 176
 - converting 179
 - creating 207
 - cropping 183
 - current 13
 - elements of 47
 - exporting 194
 - flattening 199
 - flipping 200
 - location 48
 - metadata 49
 - opening 211
 - printing 213
 - saving 176, 223
 - size 48–49
 - updating automatically 139
- duplicating 190
- Dust & Scratches filter 51

E

- elements 10
- EPS documents
 - opening 52, 103, 211
 - saving 53, 136
- equalizing 55
- event-handler 99
- EXIF 78
- exporting
 - documents 194
 - to Illustrator 75
 - to web 56
- extensions 139

F

- file extensions 139
- filter command 11, 58
 - Also see individual filters
- filters See individual filters
- filters, applying 198
- filters See individual filters
- fonts
 - Asian 141
 - English names 141
 - family 59
 - PostScript name 59
- foreground color 13

G

- galleries

- banner options 60
- color options 61
- creating 182
- layout 63
- mage options 62
- options 63
- security 65
- thumbnail options 66
- Gaussian Blur filter 67
- GIF documents
 - saving 68, 136
- Glass filter 70
- gray color 71
- grids 140
- guides 140

H

- hanging punctuation 158
- headline 78
- hex color 134
- High Pass filter 72
- highlights
 - adjusting 35
 - brightness of 89
 - threshold 89
- histograms
 - composite document 48
 - using caches in 142
- history states
 - activating 48
 - number to keep 141
 - properties 73
- HSB color 74
- hyphenation 155

I

- image pyramid 163
- images
 - black and white 162
 - compressing 163
 - inverting colors 80
 - resizing 219
- interpolation 140

J

- JavaScript
 - accessing Action Manager through 9
- JavaScript, executing from Applescript 189
- JPEG documents
 - opening 211
 - saving 81, 136
- justification 155

K

- kerning 153
- keyboard behavior 140
- keywords 78

L

- Lab color 82
- layer comps
 - applying 173
 - creating 207
 - options 85
 - recapturing 217
 - resetting 218
- layer sets
 - creating 207
 - locking contents 86
- layers
 - activating 48
 - blend mode 15
 - creating 207
 - linked 15, 16
 - linking 83, 205
 - locking contents 15-??, 15, 83
 - making visible 15
 - merging 209
 - rasterizing 216
 - sample code 17
 - styles 174
 - type of 16
 - unlinking 236
- leading 153
- left direction points 106, 107
- Lens Blur filter 89
- Lens Flare filter 91
- levels
 - adjusting 92, 172
 - automatic 19
- ligatures 156
- linking 205
- links
 - active 61
 - visited 61

M

- Maximum filter 93, 95
- Median Noise filter 94
- memory, available 13
- merging
 - channels 209
 - layers 209
- metadata 48, 78, 168
- midtones 35
- Minimum filter 95
- Motion Blur filter 97
- move command 210

N

- new features 8
- notifiers, creating 207
- NTSC colors filter 100

O

- objects

- moving 210
 - rotating 221
 - scaling 225
- Ocean Ripple filter 101
- Offset filter 102
- opacity
 - channels 31
 - fill (layers) 16
 - gamut warning 140
 - layer sets 87
 - layers 16, 83
 - picture packages 121
- open command 103
- optimizing 56
- other filters
 - Custom 40
 - High Pass 72
 - Maximum 93
 - Minimum 95
- P**
- paragraphs
 - spacing 158
 - text 155
- pasting 212
- path items *See* paths
- path points
 - defined 107
 - kind of 106
- path segments 107, 150, 151
- paths
 - closing 150
 - creating 107, 207
 - exporting to Illustrator 75
 - filling 196
 - path type 104
 - retrieving information 106
 - stroking 231
- PDF documents
 - opening 103, 108, 211
 - saving 110, 136
- PDF presentations
 - creating 182
 - formatting 128
 - looping 128
 - transition type 129
- Photo CD documents
 - opening 103, 211
- Photo CD documents, opening 113
- Photoshop documents
 - maximizing compatibility 141
 - opening 211
 - saving 118, 136
- PICT files
 - opening 211
 - saving 119, 136
- PICT resource documents
 - opening 211
 - saving 120, 136
- picture packages 121
- Pinch filter 123
- Pixar documents
 - opening 211
 - saving 124, 136
- plugins 139
- PNG documents
 - opening 211
 - saving 125, 136
- point size 141
- point text 155
- Polar Coordinates filter 126
- posterizing 127
- PostScript fonts 59
- preferences 139
- previewing 140
- printing 213
- properties 10
- PSD documents
 - maximizing compatibility 141
 - saving 118
- purging 214
- Q**
- quick mask mode 49
- R**
- Radial Blur filter 130
- rasterizing 216
- raw format documents
 - opening 103, 131, 132
 - saving 136
- RGB colors
 - defining 133
 - hex colors 134
- right direction points 106, 107
- Ripple filter 135
- ruler units 141
- S**
- save command 136, 216
 - Also see* individual document formats
- Script Events Manager 99
- scripting version 13
- selections 137
 - clearing 176
 - contracting 178
 - creating 182
 - deselecting 187
 - expanding 193
 - feathering 195
 - filling 196
 - growing 201
 - inverting 204
 - loading 206
 - selecting 226
 - storing as channel 230
 - stroking 231

- selective colors 138
- settings 139
- SGI RGB documents
 - saving 136, 143
- shadows
 - adjusting 35
 - camera raw documents 29
- Sharpen Edges filter 145
- Sharpen filter 144
- sharpen filters
 - Sharpen 144
 - Sharpen Edges 145
 - Sharpen More 146
 - Unsharp Mask 166
- Sharpen More filter 146
- Shear filter 147
- Smart Blur filter 148
- smart quotes 142
- smoothing 228
- snapshot 73
- Spherize filter 149
- straight segments (paths) 107, 150, 151
- strike thru 158

T

- Targa documents
 - saving 136, 152
- text
 - coloring 158
 - contents 153
 - converting to path 179
 - indenting 156
 - kerning 153
 - kind 155
 - orientation 159
 - picture package 121
- text composer 159
- text layers
 - content 16
 - creating 16
- Texture Fill filter 161
- threshold
 - adjusting 172
- threshold adjustments 162
- thumbnails
 - gallery images 66
 - Mac OS 141

- Windows 142
- TIFF documents
 - opening 211
 - saving 136, 163
 - saving layers in 139
- trapping 234
- trimming 235
- Twirl filter 165
- type units 142

U

- units 141, 142
- Unsharp Mask filter 166
- urgency 79
- utilities
 - Action Manager 9

V

- video alpha 142
- video filters
 - De-Interlace 41
 - NTSC colors 100
- visibility
 - channels 31
 - layer sets 87
 - layers 16, 83
- visible layers
 - merging 209

W

- warp 159
- Wave filter 167
- Web photo galleries
 - See galleries
- work paths 104
 - creating 183
 - creating from selection 208

X

- xml 168
- XMP metadata 168

Z

- Zigzag filter 169

