

**ADOBE® PHOTOSHOP® CS3**

**JAVASCRIPT SCRIPTING REFERENCE**



© 2007 Adobe Systems Incorporated. All rights reserved.

Adobe® Creative Suite® 3 Photoshop® JavaScript 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®, GoLive®, InDesign®, Illustrator®, Photoshop® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple®, Mac OS®, 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

---

<b>1</b>	<b>Introduction .....</b>	<b>31</b>
	JavaScript support in Adobe Photoshop CS3 .....	31
	Executing scripts .....	31
	Installing scripts .....	32
	Executing other scripts .....	32
	Startup scripts .....	32
	Changes Since Earlier Versions.....	32
<b>2</b>	<b>JavaScript Object Reference .....</b>	<b>35</b>
	Working with the Properties Tables .....	35
	displayDialogs.....	35
	Working with the Methods Tables .....	35
	executeAction.....	36
	Working with Method Parameters .....	36
	ActionDescriptor .....	37
	Properties.....	37
	count .....	37
	typename .....	37
	Methods .....	37
	clear .....	37
	erase .....	37
	fromStream .....	37
	getBoolean.....	37
	getClass.....	37
	getData.....	37
	getDouble .....	37
	getEnumerationType .....	37
	getEnumerationValue.....	37
	getInteger.....	37
	getKey.....	37
	getList .....	37
	getObjectType.....	38
	getObjectValue.....	38
	getPath .....	38
	getReference .....	38
	getString .....	38
	getType .....	38
	getUnitDoubleType.....	38
	getUnitDoubleValue.....	38
	hasKey.....	38
	isEqual.....	38
	putBoolean .....	38
	putClass.....	38
	putData .....	38
	putDouble .....	38
	putEnumerated .....	38

putInteger .....	38
putList .....	39
putObject .....	39
putPath .....	39
putReference .....	39
putString .....	39
putUnitDouble .....	39
toStream .....	39
ActionList .....	40
Properties .....	40
count .....	40
typename .....	40
Methods .....	40
clear .....	40
getBoolean .....	40
getClass .....	40
getData .....	40
getDouble .....	40
getEnumerationType .....	40
getEnumerationValue .....	40
getInteger .....	40
getList .....	40
getObjectType .....	40
getObjectValue .....	41
getPath .....	41
getReference .....	41
getString .....	41
getType .....	41
getUnitDoubleType .....	41
getUnitDoubleValue .....	41
putBoolean .....	41
putClass .....	41
putData .....	41
putDouble .....	41
putEnumerated .....	41
putInteger .....	41
putList .....	41
putObject .....	41
putPath .....	41
putReference .....	42
putString .....	42
putUnitDouble .....	42
ActionReference .....	43
Properties .....	43
typename .....	43
Methods .....	43
getContainer .....	43
getDesiredClass .....	43
getEnumeratedType .....	43
getEnumeratedValue .....	43
getForm .....	43

getIdentifier .....	43
getIndex .....	43
getName .....	43
getOffset .....	43
getProperty .....	43
putClass .....	43
putEnumerated .....	44
putIdentifier .....	44
putIndex .....	44
putName .....	44
putOffset .....	44
putProperty .....	44
Application .....	45
Properties .....	45
activeDocument .....	45
backgroundColor .....	45
colorSettings .....	45
displayDialogs .....	45
documents .....	45
fonts .....	45
foregroundColor .....	45
freeMemory .....	45
locale .....	45
macintoshFileTypes .....	45
measurementLog .....	45
name .....	46
notifiers .....	46
notifiersEnabled .....	46
path .....	46
playbackDisplayDialogs .....	46
playbackParameters .....	46
preferences .....	46
preferencesFolder .....	46
recentFiles .....	46
scriptingBuildDate .....	46
scriptingVersion .....	46
typename .....	46
version .....	46
windowsFileTypes .....	46
Methods .....	47
batch .....	47
beep .....	47
bringToFront .....	47
charIDToTypeID .....	47
doAction .....	47
eraseCustomOptions .....	47
executeAction .....	47
executeActionGet .....	47
featureEnabled .....	48
getCustomOptions .....	48
load .....	48

makeContactSheet.....	48
makePDFPresentation .....	48
makePhotoGallery.....	48
makePhotomerge.....	48
makePicturePackage .....	48
open .....	49
openDialog .....	49
purge.....	49
putCustomOptions .....	49
refresh .....	49
stringIDToTypeID.....	49
TypeIDToCharID .....	50
TypeIDToStringID .....	50
ArtLayer.....	53
Properties.....	53
allLocked.....	53
blendMode.....	53
bounds.....	53
fillOpacity.....	53
grouped .....	53
isBackgroundLayer.....	53
kind .....	53
linkedLayers.....	54
name.....	54
opacity .....	54
parent.....	54
pixelsLocked .....	54
positionLocked .....	54
textItem.....	54
transparentPixelsLocked.....	54
typename .....	54
visible .....	54
Methods .....	54
adjustBrightnessContrast .....	54
adjustColorBalance .....	54
adjustCurves .....	55
adjustLevels .....	55
applyAddNoise .....	55
applyAverage .....	55
applyBlur.....	55
applyBlurMore .....	55
applyClouds.....	55
applyCustomFilter .....	55
applyDeInterlace.....	55
applyDespeckle .....	55
applyDifferenceClouds.....	55
applyDiffuseGlow .....	55
applyDisplace.....	56
applyDustAndScratches.....	56
applyGaussianBlur.....	56
applyGlassEffect.....	56

applyHighPass .....	56
applyLensBlur .....	57
applyLensFlare.....	57
applyMaximum .....	58
applyMedianNoise .....	58
applyMinimum .....	58
applyMotionBlur .....	58
applyNTSC.....	58
applyOceanRipple .....	58
applyOffset.....	58
applyPinch .....	58
applyPolarCoordinates.....	58
applyRadialBlur .....	58
applyRipple .....	58
applySharpen.....	58
applySharpenEdges.....	59
applySharpenMore .....	59
applyShear .....	59
applySmartBlur.....	59
applySpherize .....	59
applyStyle .....	59
applyTextureFill.....	59
applyTwirl.....	59
applyUnSharpMask.....	59
applyWave.....	59
applyZigZag.....	60
autoContrast.....	60
autoLevels .....	60
clear .....	60
copy .....	60
cut.....	60
desaturate .....	60
duplicate .....	60
equalize .....	60
invert .....	60
link.....	60
merge.....	61
mixChannels .....	61
move.....	61
photoFilter .....	62
posterize .....	62
rasterize.....	62
remove.....	62
resize .....	62
rotate.....	62
selectiveColor.....	62
shadowHighlight .....	62
threshold.....	63
translate .....	63
unlink .....	63
ArtLayers.....	65

Properties.....	65
length.....	65
parent.....	65
typename .....	65
Methods .....	65
index.....	65
add .....	65
getByName .....	65
removeAll .....	65
BatchOptions .....	66
Properties.....	66
destination .....	66
destinationFolder .....	66
errorFile .....	66
fileNaming.....	66
macintoshCompatible .....	66
overrideOpen.....	66
overrideSave.....	67
startingSerial .....	67
suppressOpen.....	67
suppressProfile .....	67
typename .....	67
unixCompatible.....	67
windowsCompatible .....	67
BitmapConversionOptions .....	68
Properties.....	68
angle.....	68
frequency.....	68
method.....	68
patternName .....	68
resolution.....	68
shape.....	68
typename .....	68
BMPSaveOptions .....	69
Properties.....	69
alphaChannels .....	69
depth.....	69
flipRowOrder .....	69
osType.....	69
rleCompression .....	69
typename .....	69
CameraRAWOpenOptions .....	70
Properties.....	70
bitsPerChannel .....	70
blueHue.....	70
blueSaturation .....	70
brightness.....	70
chromaticAberrationBY .....	70
chromaticAberrationRC.....	70
colorNoiseReduction .....	70
colorSpace.....	70



contrast .....	70
exposure .....	70
greenHue.....	70
greenSaturation .....	70
luminanceSmoothing .....	70
redHue .....	70
redSaturation .....	70
resolution.....	70
saturation .....	70
settings.....	70
shadows .....	70
shadowTint .....	71
sharpness.....	71
size .....	71
temperature.....	71
tint.....	71
typename .....	71
vignettingAmount .....	71
vignettingMidpoint .....	71
whiteBalance.....	71
Channel .....	72
Properties.....	72
color.....	72
histogram .....	72
kind .....	72
name.....	72
opacity .....	72
parent.....	72
typename .....	73
visible .....	73
Methods .....	73
duplicate .....	73
merge.....	73
remove.....	73
Channels .....	74
Properties.....	74
length.....	74
parent.....	74
typename .....	74
Methods .....	74
index.....	74
add .....	74
getByName .....	74
removeAll .....	74
CMYKColor .....	79
Properties.....	79
black .....	79
cyan.....	79
magenta.....	79
typename .....	79
yellow.....	79

ColorSampler .....	80
Properties.....	80
color.....	80
position .....	80
parent.....	80
typename .....	80
Methods .....	80
move.....	80
remove.....	80
ColorSamplers .....	81
Properties.....	81
length.....	81
parent.....	81
typename .....	81
Methods .....	81
index.....	81
add .....	81
getByName .....	81
removeAll .....	81
ContactSheetOptions .....	82
Properties.....	82
acrossFirst.....	82
bestFit .....	82
caption.....	82
columnCount .....	82
flatten.....	82
font.....	82
fontSize.....	82
height.....	82
horizontal .....	82
mode .....	82
resolution.....	82
rowCount.....	82
typename .....	82
useAutoSpacing.....	82
vertical .....	83
width .....	83
CountItem .....	84
Properties.....	84
position .....	84
parent.....	84
typename .....	84
Methods .....	84
remove.....	84
CountItems .....	85
Properties.....	85
length.....	85
parent.....	85
typename .....	85
Methods .....	85
index.....	85

add .....	85
getByName .....	85
removeAll .....	85
DCS1_SaveOptions .....	86
Properties.....	86
dCS .....	86
embedColorProfile .....	86
encoding.....	86
halftoneScreen .....	86
interpolation.....	86
preview.....	86
transferFunction.....	86
typename .....	86
vectorData.....	86
DCS2_SaveOptions .....	87
Properties.....	87
dCS .....	87
embedColorProfile .....	87
encoding.....	87
halftoneScreen .....	87
interpolation.....	87
multiFileDCS .....	87
preview.....	87
spotColors .....	87
transferFunction.....	87
typename .....	87
vectorData.....	87
DICOMOpenOptions .....	88
Properties.....	88
anonymize.....	88
columns.....	88
reverse .....	88
rows .....	88
showOverlays.....	88
typename .....	88
windowLevel .....	88
windowWidth .....	88
Document .....	89
Properties.....	89
activeChannels .....	89
activeHistoryBrushSource .....	89
activeHistoryState .....	89
activeLayer .....	89
artLayers.....	89
backgroundLayer.....	89
bitsPerChannel .....	89
channels.....	89
colorProfileName.....	89
colorProfileType .....	89
colorSamplers .....	90
componentChannels.....	90

countItems .....	90
fullName .....	90
height .....	90
histogram .....	90
historyStates .....	90
info .....	90
layerComps .....	90
layers .....	90
layerSets .....	90
managed .....	90
measurementScale .....	90
mode .....	90
name .....	90
parent .....	91
path .....	91
pathItems .....	91
pixelAspectRatio .....	91
quickMaskMode .....	91
resolution .....	91
saved .....	91
selection .....	91
typename .....	91
width .....	91
xmpMetadata .....	91
Methods .....	92
autoCount .....	92
changeMode .....	92
close .....	92
convertProfile .....	92
crop .....	93
duplicate .....	93
exportDocument .....	93
flatten .....	93
flipCanvas .....	93
importAnnotations .....	93
mergeVisibleLayers .....	93
paste .....	94
print .....	94
rasterizeAllLayers .....	94
recordMeasurements .....	94
resizeCanvas .....	94
resizeImage .....	94
revealAll .....	95
rotateCanvas .....	95
save .....	95
saveAs .....	95
splitChannels .....	95
suspendHistory .....	95
trap .....	95
trim .....	96
DocumentInfo .....	98

Properties.....	98
author .....	98
authorPosition .....	98
caption.....	98
captionWriter .....	98
category .....	98
city.....	98
copyrighted .....	98
copyrightNotice .....	98
country .....	98
creationDate.....	98
credit .....	98
exif.....	98
headline .....	98
instructions .....	98
jobName .....	98
keywords.....	98
ownerUrl .....	99
parent.....	99
provinceState.....	99
source.....	99
supplementalCategories.....	99
title .....	99
transmissionReference .....	99
typename .....	99
urgency .....	99
Documents .....	102
Properties.....	102
length.....	102
parent.....	102
typename .....	102
Methods .....	102
index.....	102
add .....	102
getByName .....	102
EPSSaveOptions.....	103
Properties.....	103
antiAlias.....	103
constrainProportions .....	103
height.....	103
mode.....	103
resolution.....	103
typename .....	103
width .....	103
EPSSaveOptions.....	104
Properties.....	104
embedColorProfile.....	104
encoding.....	104
halftoneScreen .....	104
interpolation.....	104
preview .....	104

psColorManagement .....	104
transferFunction.....	104
transparentWhites.....	104
typename .....	104
vectorData.....	104
ExportOptionsIllustrator .....	105
Properties.....	105
path.....	105
pathName.....	105
typename .....	105
ExportOptionsSaveForWeb .....	106
Properties.....	106
blur .....	106
colorReduction .....	106
colors.....	106
dither.....	106
ditherAmount .....	106
format .....	106
includeProfile .....	106
interlaced.....	106
lossy .....	106
matteColor .....	106
optimized .....	106
PNG8.....	107
quality .....	107
transparency.....	107
transparencyAmount.....	107
transparencyDither .....	107
typename .....	107
webSnap .....	107
GalleryBannerOptions .....	108
Properties.....	108
contactInfo.....	108
date.....	108
font.....	108
fontSize.....	108
photographer.....	108
siteName .....	108
typename .....	108
GalleryCustomColorOptions .....	109
Properties.....	109
activeLinkColor.....	109
backgroundColor.....	109
bannerColor.....	109
linkColor .....	109
textColor .....	109
typename .....	109
visitedLinkColor.....	109
GalleryImagesOptions .....	110
Properties.....	110
border .....	110

caption.....	110
dimension.....	110
font.....	110
fontSize.....	110
imageQuality.....	110
includeCopyright.....	110
includeCredits.....	110
includeFilename.....	111
includeTitle .....	111
numericLinks.....	111
resizeConstraint .....	111
resizeImages.....	111
typename .....	111
GalleryOptions.....	112
Properties.....	112
addSizeAttributes .....	112
bannerOptions .....	112
customColorOptions .....	112
emailAddress.....	112
imagesOptions .....	112
includeSubFolders .....	112
layoutStyle .....	112
preserveAllMetadata .....	112
securityOptions .....	112
thumbnailOptions.....	112
typename .....	112
useShortExtension .....	112
useUTF8Encoding .....	112
GallerySecurityOptions .....	113
Properties.....	113
content .....	113
font.....	113
fontSize.....	113
opacity .....	113
text .....	113
textColor .....	113
textPosition.....	113
textRotate .....	113
typename .....	113
GalleryThumbnailOptions.....	114
Properties.....	114
border .....	114
caption.....	114
columnCount .....	114
dimension.....	114
font.....	114
fontSize.....	114
includeCopyright.....	114
includeCredits.....	114
includeFilename.....	114
includeTitle .....	114

rowCount.....	114
size .....	114
typename .....	114
GIFSaveOptions.....	115
Properties.....	115
colors.....	115
dither.....	115
ditherAmount .....	115
forced .....	115
interlaced.....	115
matte.....	115
palette.....	115
preserveExactColors .....	115
transparency.....	116
typename .....	116
GrayColor.....	117
Properties.....	117
gray .....	117
typename .....	117
HistoryState .....	118
Properties.....	118
name.....	118
parent.....	118
snapshot .....	118
typename .....	118
HistoryStates .....	119
Properties.....	119
length.....	119
parent.....	119
typename .....	119
Methods .....	119
index.....	119
getByName .....	119
HSBColor.....	120
Properties.....	120
brightness.....	120
hue .....	120
saturation .....	120
typename .....	120
IndexedConversionOptions .....	121
Properties.....	121
colors.....	121
dither.....	121
ditherAmount .....	121
forced .....	121
matte.....	121
palette.....	121
preserveExactColors .....	121
transparency.....	121
typename .....	121
JPEGSaveOptions .....	122



Properties.....	122
embedColorProfile.....	122
formatOptions .....	122
matte.....	122
quality .....	122
scans .....	122
typename .....	122
LabColor .....	123
Properties.....	123
a.....	123
b .....	123
l.....	123
typename .....	123
LayerComp.....	124
Properties.....	124
appearance .....	124
comment .....	124
name.....	124
parent.....	124
position .....	124
selected .....	124
typename .....	124
visibility .....	124
Methods .....	124
apply.....	124
recapture .....	124
remove.....	125
resetfromComp .....	125
LayerComps.....	126
Properties.....	126
length.....	126
parent.....	126
typename .....	126
Methods .....	126
index.....	126
add .....	126
getByName .....	126
removeAll .....	126
Layers.....	127
Properties.....	127
length.....	127
parent.....	127
typename .....	127
Methods .....	127
index.....	127
getByName .....	127
removeAll .....	127
LayerSet .....	128
Properties.....	128
allLocked.....	128
artLayers.....	128

blendMode.....	128
bounds.....	128
enabledChannels.....	128
layers.....	128
layerSets.....	128
linkedLayers.....	128
name.....	128
opacity.....	128
parent.....	128
typename.....	128
visible.....	128
Methods.....	129
duplicate.....	129
link.....	129
merge.....	129
move.....	129
remove.....	129
resize.....	129
rotate.....	129
translate.....	129
unlink.....	129
LayerSets.....	130
Properties.....	130
length.....	130
parent.....	130
typename.....	130
Methods.....	130
index.....	130
add.....	130
getByName.....	130
removeAll.....	130
MeasurementLog.....	132
Methods.....	132
exportMeasurements.....	132
deleteMeasurements.....	132
MeasurementScale.....	133
Properties.....	133
pixelLength.....	133
logicalLength.....	133
logicalUnits.....	133
NoColor.....	134
Properties.....	134
typename.....	134
Notifier.....	135
Properties.....	135
event.....	135
eventClass.....	135
eventFile.....	135
parent.....	135
typename.....	135
Methods.....	136

remove.....	136
Notifiers.....	137
Properties.....	137
length.....	137
parent.....	137
typename .....	137
Methods .....	138
index.....	138
add .....	138
removeAll .....	138
PathItem .....	139
Properties.....	139
kind .....	139
name.....	139
parent.....	139
SubPathItems.....	139
typename .....	139
Methods .....	139
deselect .....	139
duplicate .....	139
fillPath.....	139
makeClippingPath.....	140
makeSelection .....	140
remove.....	140
select .....	140
strokePath .....	140
PathItems .....	143
Properties.....	143
length.....	143
parent.....	143
typename .....	143
Methods .....	143
index.....	143
add .....	143
getByName .....	143
removeAll .....	143
PathPoint.....	144
Properties.....	144
anchor.....	144
kind .....	144
leftDirection.....	144
parent.....	144
rightDirection.....	144
typename .....	144
PathPointInfo .....	145
Properties.....	145
anchor.....	145
kind .....	145
leftDirection.....	145
rightDirection.....	145
typename .....	145

PathPoints.....	146
Properties.....	146
length.....	146
parent.....	146
typename .....	146
Methods .....	146
index.....	146
PDFOpenOptions .....	147
Properties.....	147
antiAlias.....	147
bitsPerChannel .....	147
constrainProportions .....	147
cropPage.....	147
height.....	147
mode.....	147
name.....	147
page.....	147
resolution.....	147
suppressWarnings .....	147
typename .....	147
usePageNumber .....	147
width .....	147
PDFSaveOptions .....	148
Properties.....	148
alphaChannels .....	148
annotations.....	148
colorConversion.....	148
convertToEightBit.....	148
description .....	148
destinationProfile .....	148
downgradeColorProfile .....	148
downSample .....	148
downSampleSize .....	148
downSampleSizeLimit .....	148
embedColorProfile.....	148
embedFonts .....	148
embedThumbnail.....	148
encoding.....	148
interpolation.....	148
jpegQuality .....	149
layers .....	149
optimizeForWeb .....	149
outputCondition .....	149
outputConditionID.....	149
PDFCompatibility.....	149
PDFStandard .....	149
preserveEditing .....	149
presetFile .....	149
profileInclusionPolicy.....	149
registryName.....	149
spotColors .....	149

tileSize.....	149
transparency.....	149
typename .....	149
useOutlines .....	149
vectorData.....	150
view.....	150
PhotoCDOpenOptions .....	151
Properties.....	151
colorProfileName.....	151
colorSpace.....	151
orientation.....	151
pixelSize .....	151
resolution.....	151
typename .....	151
PhotoshopSaveOptions.....	152
Properties.....	152
alphaChannels .....	152
annotations.....	152
embedColorProfile.....	152
layers .....	152
spotColors .....	152
typename .....	152
PICTFileSaveOptions .....	153
Properties.....	153
alphaChannels .....	153
compression .....	153
embedColorProfile.....	153
resolution.....	153
typename .....	153
PICTResourceSaveOptions.....	154
Properties.....	154
alphaChannels .....	154
compression .....	154
embedColorProfile.....	154
name.....	154
resolution.....	154
resourceID .....	154
typename .....	154
PicturePackageOptions.....	155
Properties.....	155
content .....	155
flatten.....	155
font.....	155
fontSize.....	155
layout .....	155
mode .....	155
opacity .....	155
resolution.....	155
text .....	155
textColor .....	155
textPosition.....	155

textRotate .....	155
typename .....	155
PixarSaveOptions .....	156
Properties.....	156
alphaChannels .....	156
typename .....	156
PNGSaveOptions .....	157
Properties.....	157
interlaced.....	157
typename .....	157
Preferences .....	158
Properties.....	158
additionalPluginFolder.....	158
appendExtension.....	158
askBeforeSavingLayeredTIFF.....	158
autoUpdateOpenDocuments .....	158
beepWhenDone.....	158
colorChannelsInColor .....	158
colorPicker.....	158
columnGutter.....	158
columnWidth .....	159
createFirstSnapshot.....	159
dynamicColorSliders.....	159
editLogItems .....	159
exportClipboard.....	159
fontPreviewSize.....	159
fullSizePreview.....	159
gamutWarningOpacity.....	159
gridSize .....	159
gridStyle .....	159
gridSubDivisions .....	159
guideStyle.....	159
iconPreview .....	159
imageCacheLevels .....	159
imagePreviews .....	159
interpolation.....	159
keyboardZoomResizesWindows.....	160
macOSThumbnail .....	160
maximizeCompatibility .....	160
maxRAMuse.....	160
nonLinearHistory .....	160
numberOfHistoryStates .....	160
otherCursors .....	160
paintingCursors.....	160
parent.....	160
pixelDoubling .....	160
pointSize .....	160
recentFileListLength.....	160
rulerUnits .....	160
saveLogItems .....	160
saveLogItemsFile .....	160

savePaletteLocations .....	161
showAsianTextOptions .....	161
showEnglishFontNames .....	161
showSliceNumber .....	161
showToolTips .....	161
smartQuotes .....	161
typename .....	161
typeUnits .....	161
useAdditionalPluginFolder .....	161
useHistoryLog .....	161
useLowerCaseExtension .....	161
useShiftKeyForToolSwitch .....	161
useVideoAlpha .....	161
windowsThumbnail .....	161
PresentationOptions .....	162
Properties .....	162
autoAdvance .....	162
includeFilename .....	162
interval .....	162
loop .....	162
magnification .....	162
pdfFileOptions .....	162
presentation .....	162
transition .....	162
typename .....	162
RawFormatOpenOptions .....	163
Properties .....	163
bitsPerChannel .....	163
byteOrder .....	163
channelNumber .....	163
headerSize .....	163
height .....	163
interleaveChannels .....	163
retainHeader .....	163
typename .....	163
width .....	163
RawSaveOptions .....	164
Properties .....	164
alphaChannels .....	164
spotColors .....	164
typename .....	164
RGBColor .....	165
Properties .....	165
blue .....	165
green .....	165
hexValue .....	165
red .....	165
typename .....	165
Selection .....	166
Properties .....	166
bounds .....	166

parent.....	166
solid .....	166
typename .....	166
Methods .....	166
clear .....	166
contract .....	166
copy .....	166
cut.....	166
deselect .....	166
expand.....	166
feather.....	166
fill .....	167
grow .....	167
invert .....	167
load .....	167
makeWorkPath .....	167
resize .....	167
resizeBoundary .....	167
rotate.....	167
rotateBoundary .....	167
select .....	167
selectAll.....	167
selectBorder.....	168
similar.....	168
smooth .....	168
store.....	168
stroke.....	168
translate .....	168
translateBoundary .....	168
SGIRGBSaveOptions .....	171
Properties.....	171
alphaChannels .....	171
spotColors .....	171
typename .....	171
SolidColor.....	172
Properties.....	172
cmyk .....	172
gray .....	172
hsb.....	172
lab.....	172
model.....	172
nearestWebColor .....	172
rgb.....	172
typename .....	172
Methods .....	172
isEqual.....	172
SubPathInfo.....	173
Properties.....	173
closed .....	173
entireSubPath .....	173
operation .....	173



typename .....	173
SubPathItem.....	174
Properties.....	174
closed.....	174
operation .....	174
parent.....	174
pathPoints .....	174
typename .....	174
SubPathItems.....	175
Properties.....	175
length.....	175
parent.....	175
typename .....	175
Methods .....	175
index.....	175
TargaSaveOptions.....	176
Properties.....	176
alphaChannels .....	176
resolution.....	176
rleCompression .....	176
typename .....	176
TextFont.....	177
Properties.....	177
family.....	177
name.....	177
parent.....	177
postScriptName .....	177
style.....	177
typename .....	177
TextFonts.....	178
Properties.....	178
length.....	178
parent.....	178
typename .....	178
Methods .....	178
index.....	178
getByName .....	178
TextItem.....	179
Properties.....	179
alternateLigatures .....	179
antiAliasMethod.....	179
autoKerning.....	179
autoLeadingAmount.....	179
baselineShift .....	179
capitalization .....	179
color.....	179
contents .....	179
desiredGlyphScaling.....	179
desiredLetterScaling.....	180
desiredWordScaling .....	180
direction .....	180

fauxBold .....	180
fauxItalic .....	180
firstLineIndent .....	180
font .....	180
hangingPunctuation .....	180
height .....	181
horizontalScale .....	181
hyphenateAfterFirst .....	181
hyphenateBeforeLast .....	181
hyphenateCapitalWords .....	181
hyphenateWordsLongerThan .....	181
hyphenation .....	181
hyphenationZone .....	181
hyphenLimit .....	181
justification .....	181
kind .....	181
language .....	181
leading .....	181
leftIndent .....	181
ligatures .....	181
maximumGlyphScaling .....	182
maximumLetterScaling .....	182
maximumWordScaling .....	182
minimumGlyphScaling .....	183
minimumLetterScaling .....	183
minimumWordScaling .....	183
noBreak .....	183
oldStyle .....	183
parent .....	184
position .....	184
rightIndent .....	184
size .....	184
spaceAfter .....	184
spaceBefore .....	184
strikeThru .....	184
textComposer .....	184
tracking .....	184
typename .....	184
underline .....	184
useAutoLeading .....	184
verticalScale .....	184
warpBend .....	184
warpDirection .....	184
warpHorizontalDistortion .....	185
warpStyle .....	185
warpVerticalDistortion .....	185
width .....	185
Methods .....	185
convertToShape .....	185
createPath .....	185
TiffSaveOptions .....	186

Properties.....	186
alphaChannels.....	186
annotations.....	186
byteOrder.....	186
embedColorProfile.....	186
imageCompression.....	186
interleaveChannels.....	186
jpegQuality.....	186
layerCompression.....	186
layers.....	186
saveImagePyramid.....	186
spotColors.....	186
transparency.....	186
typename.....	186
xmpMetadata.....	187
Properties.....	187
parent.....	187
rawData.....	187
typename.....	187

### **3 Scripting Constants ..... 188**

AdjustmentReference.....	188
AnchorPosition.....	188
AntiAlias.....	188
AutoKernType.....	188
BatchDestinationType.....	188
BitmapConversionType.....	188
BitmapHalfToneType.....	188
BitsPerChannelType.....	188
BlendMode.....	189
BMPDepthType.....	189
ByteOrder.....	189
CameraRAWSettingsType.....	189
CameraRAWSize.....	189
ChangeMode.....	190
ChannelType.....	190
ColorBlendMode.....	190
ColorModel.....	190
ColorPicker.....	190
ColorProfile.....	190
ColorReductionType.....	191
ColorSpaceType.....	191
CopyrightedType.....	191
CreateFields.....	191
CropToType.....	191
DCSType.....	191
DepthMapSource.....	191
DescValueType.....	191
DialogModes.....	191
Direction.....	191
DisplacementMapType.....	192

Dither .....	192
DocumentFill .....	192
DocumentMode .....	192
EditLogItemsType .....	192
ElementPlacement .....	192
EliminateFields .....	192
ExportType .....	192
Extension .....	192
FileNamingType .....	193
FontPreviewType .....	193
ForcedColors .....	193
FormatOptions .....	193
GalleryConstrainType .....	193
GalleryFontType .....	193
GallerySecurityTextColorType .....	193
GallerySecurityTextPositionType .....	193
GallerySecurityTextRotateType .....	194
GallerySecurityType .....	194
GalleryThumbSizeType .....	194
Geometry .....	194
GridLineStyle .....	194
GridSize .....	194
GuideLineStyle .....	194
IllustratorPathType .....	194
Intent .....	194
JavaScriptExecutionMode .....	194
Justification .....	194
Language .....	195
LayerCompression .....	195
LayerKind .....	195
LensType .....	195
MagnificationType .....	195
MatteType .....	195
MeasurementRange .....	195
MeasurementSource .....	196
NewDocumentMode .....	196
NoiseDistribution .....	196
OffsetUndefinedAreas .....	196
OpenDocumentMode .....	196
OpenDocumentType .....	196
OperatingSystem .....	196
Orientation .....	196
OtherPaintingCursors .....	197
PaintingCursors .....	197
Palette .....	197
PathKind .....	197
PDFCompatibility .....	197
PDFEncoding .....	197
PDFResample .....	197
PDFStandard .....	198
PhotoCDColorSpace .....	198

PhotoCDSize .....	198
PICTBitsPerPixels .....	198
PICTCompression .....	198
PicturePackageTextType .....	198
PointKind .....	198
PointType .....	198
PolarConversionType .....	199
Preview .....	199
PrintEncoding .....	199
PurgeTarget .....	199
QueryStateType .....	199
RadialBlurMethod .....	199
RadialBlurQuality .....	199
RasterizeType .....	199
ReferenceFormType .....	199
ResampleMethod .....	199
ResetTarget .....	199
RippleSize .....	199
SaveBehavior .....	199
SaveDocumentType .....	200
SaveEncoding .....	200
SaveLogItemsType .....	200
SaveOptions .....	200
SelectionType .....	200
ShapeOperation .....	200
SmartBlurMode .....	200
SmartBlurQuality .....	201
SourceSpaceType .....	201
SpherizeMode .....	201
StrikeThruType .....	201
StrokeLocation .....	201
TargaBitsPerPixels .....	201
TextCase .....	201
TextComposer .....	201
TextType .....	201
TextureType .....	201
TIFFEncoding .....	201
ToolType .....	202
TransitionType .....	202
TrimType .....	202
TypeUnits .....	202
UndefinedAreas .....	202
UnderlineType .....	202
Units .....	203
Urgency .....	203
WarpStyle .....	203
WaveType .....	203
WhiteBalanceType .....	203
ZigZagType .....	203

## **4 JavaScript Resource.....204**

JavaScript Resource Syntax .....	204
Basic JavaScript Resource Example .....	205
Enable Info Grammar .....	205
Undefined Values in Enable Info Evaluation .....	207
Using the “in” Function .....	208
Action Manager Automation .....	208
Terminology Dictionary .....	208
Value Type Definitions .....	209
Uniqueness Rules for Terminology Entries .....	210
Terminology Definition Example .....	210
<b>Appendix A: Event ID Codes .....</b>	<b>212</b>
<b>Index .....</b>	<b>220</b>

This reference describes the objects and methods in the Adobe® Photoshop® CS® 3 JavaScript™ type library. A companion document, *Photoshop CS3 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 JavaScript-specific features.

Adobe Photoshop CS3 uses ExtendScript, Adobe's extended implementation of JavaScript. See [JavaScript support in Adobe Photoshop CS3](#) for additional information.

This book contains the following sections:

- This introduction, which describes scripting support in Adobe Photoshop CS3, and lists changes to the JavaScript interface since the previous release.
- [JavaScript Object Reference](#), which provides a complete reference for all Photoshop DOM objects and commands.
- [Scripting Constants](#), which lists all enumerations used in the Photoshop type library.

## JavaScript support in Adobe Photoshop CS3

For a JavaScript file to be recognized by Photoshop as a valid script file, it must use either a `.js` or a `.jsx` extension.

On the Mac OS, there is no difference in the way scripts with the two extensions function. On Windows, if the script files is opened from inside Photoshop, there is no difference between using the `.js` and `.jsx` extension. However, if the script is launched by double-clicking on it, a script with the `.js` extension is interpreted with the Microsoft JScript engine, and it cannot launch Adobe Photoshop CS3. For Windows, using the `.jsx` extension is preferable, since it interprets the script with the ExtendScript engine.

All of the Adobe Creative Suite 3 applications, including Adobe Photoshop CS3, use ExtendScript, Adobe's extended implementation of JavaScript. ExtendScript files are distinguished by the `.jsx` extension. ExtendScript offers all standard JavaScript features, plus additional features and utilities, such as:

- A debugging environment (the ExtendScript Toolkit)
- A localization utility
- Tools that allow you to combine scripts and direct them to particular applications
- Platform-independent file and folder representation

Many of the JavaScript objects and methods use objects defined in ExtendScript, such as the `File` object, the `Folder` object, and the `UnitValue` object. For that reason, using the `.jsx` extension for your script files is preferable. For details of these and additional features, see the *JavaScript Tools Guide CS3*.

## Executing scripts

The Adobe Photoshop CS3 interface includes a Scripts menu (**File > Scripts**) which provides quick and easy access to your JavaScripts. Scripts can be listed directly as menu items that run when you select them, or you can navigate to and run any JavaScript in your file system.

If Adobe Photoshop CS3 encounters an error during script execution, it displays the error message.

## Installing scripts

To install a JavaScript in the Scripts menu, place it in the Scripts folder (**Photoshop CS3/Presets/Scripts**). The names of the scripts in the Scripts folder, without the file name extension, will be displayed in the Scripts menu. Any number of scripts may be installed in the Scripts menu.

Scripts added to the Scripts folder while Adobe Photoshop CS3 is running will not appear in the Scripts menu until the next time you launch the application.

All scripts found in the Scripts folder and sub-folders are displayed at the top level of the **File > Scripts** menu. The addition of sub-folders does not add a hierarchical organization to the Scripts menu.

## Executing other scripts

The **Browse** item at the end of the **Scripts** menu (**File > Scripts > Browse**) allows you to execute scripts which are not installed in the Scripts folder. You can also use Browse to select scripts installed in the Scripts folder after the application was last launched.

Selecting **Browse** displays a file browser dialog which allows you to select a script file for execution. Only `.js` or `.jsx` files are displayed in the browse dialog. When you select a script file, it is executed the same way as an installed script.

## Startup scripts

On startup, Adobe Photoshop CS3 executes all `.jsx` files that it finds in the startup folders.

- On Windows, the startup folder for user-defined scripts is:

```
C:\Program Files\Common Files\Adobe\Startup Scripts CS3\Adobe Photoshop
```

- On Mac OS, the startup folder for user-defined scripts is:

```
~/Library/Application Support/Adobe/Startup Scripts CS3/Adobe Photoshop
```

If a script is meant to be executed only by Adobe Photoshop CS3, it must include code such as the following:

```
if( BridgeTalk.appName == "photoshop" ) {  
    //continue executing script  
}
```

For additional details, see the *JavaScript Tools Guide CS3*.

## Changes Since Earlier Versions

The following changes have been made to the JavaScript object model and language support in Adobe Photoshop CS3:

- Documentation Changes:
  - Documentation for ExtendScript objects (such as File and Folder objects, Script UI, and the Dollar (\$) object) is now found in a separate manual entitled the *JavaScript Tools Guide CS3*.

- On Windows, the document can be found in the following folder:

```
C:\Program Files\Adobe\Adobe Utilities\ExtendScript Toolkit 2\SDK
```



- On Mac OS, the document can be found in the following folder:

Applications/Utilities/Adobe Utilities/ExtendScript Toolkit 2/SDK

- Documentation that provides an overview of Scripting is now found in a new document entitled *Introduction to Scripting*, instead of in the *Photoshop CS3 Scripting Guide*.
- Photoshop CS3 has an Extended Version and a Standard Version. Some additional features are available in the Extended Version.
- `CountItems` provides support in scripting for the Count Tool. A new collection, `CountItems`, is defined, with a corresponding `countItems` property on `Document`, that refers to the collection `CountItems`. This feature is available only in the Extended Version.
- `ColorSamplers` have been provided for the `Document` object. A new collection, `ColorSamplers`, is defined, with a corresponding `colorSamplers` property on `Document` that refers to the collection `ColorSamplers`.
- `DICOMOpenOptions`: a new class that provides options for opening files in the DICOM format; this feature is only available in the Extended Version.
- The measurement scale feature adds a `measurementScale` property to the `Document` object, which accesses the `MeasurementScale` object. The `measurementScale` property on `Document` is read-only (i.e. you cannot create a new `MeasurementScale` object), but the properties of the `MeasurementScale` object (`pixelLength`, `logicalLength`, and `logicalUnits`) can be set directly. This feature is available only in the Extended Version.
- New JavaScript Resource that allows scripts to behave like plug-ins. Provides a way to specify a menu the script appears in, a terminology resource to allow the script to function with the Action Manager, and a way to indicate whether the scripts is enabled in the menu or not.
- New and modified methods:
  - `Application.featureEnabled()`: Allows you to determine if a feature with a given name is enabled, as related to the Extended version of Photoshop.
  - `Application.refresh()`: Pauses the script while the application refreshes.
  - `Application.openDialog()`: Allows you to use the Photosho open dialog to select files.
  - `Application.putCustomOptions()`, `Application.getCustomOptions()`, `Application.eraseCustomOptions()`: Allow you to save and load your parameters
  - `Application.open()`: Includes a boolean parameter `smartObject`, which indicates whether to create a smart object around the document opened.
  - `Document.duplicate()`: Provides new parameters for naming the duplciated document and setting a "merge visible layers" only option.
  - `Documents.add()`: Provides new parameters for document depth and color profile name.
  - `Document.suspendHistory()`: Allows you to provide a JavaScript string that executes as a single history state. Provides one undo and history item for multiple changes to a document.
- New properties:
  - `scriptingBuildDate` added to the `Application` object.
  - `recentFiles` added to the `Application` object.
  - `Solid`, boolean, added to `Selection` object, indicates if the bounding rectangle is solid.
- Modified enumerations:
  - `HARDMIX` has been added to the `BlendMode` enumeration.
  - `HARDMIXBLEND` has been added to the `ColorBlendMode` enumeration.

- `PHOTOFILTER` and `EXPOSURE` have been added to the `LayerKind` enumeration.
- `DICOM` has been added to the `OpenDocumentType` enumeration.

#### Bug fixes

- Fixed `bounds` property on `Selection` object.
- Fixed `PathItem.duplicate()` method, it now returns the `PathItem`.
- The `Document.resizeImage` command now behaves correctly for percentages. In CS2, it was off by 2 decimal places. (E.g 25% = 0.25)
- The `LensBlurOptions` object has been removed. All options for the Lens Blur filter now appear directly as parameters to the `ArtLayer.applyLensBlur()` method.
- Preferences properties `macOSTumbnail` and `windowsTumbnail` have been revised to correct the spelling error, to `macOSThumbnail` and `windowsThumbnail`.
- Fixed `CameraRAWOpenOptions`.

The objects of CameraRAW, the JavaScript type library for Adobe® Photoshop® CS3, are presented alphabetically and in tabular format in this chapter.

Object properties and methods are described in separate tables for each object. See [Working with the Properties Tables](#) and [Working with the Methods Tables](#) for information on how to use these tables.

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

## Working with the Properties Tables

The Properties table for an object lists the following:

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

When the value type is a constant or another object, the value is a hypertext link to the constant's or object's listing, as in the following Properties table sample.

- The property's input status: read-only or read-write.
- A description that explains what the property is

Descriptions are omitted for self-explanatory properties.

Property	Value Type	What it is
<code>displayDialogs</code>	<a href="#">DialogModes</a>	Read-write. Controls whether or not Adobe Photoshop CS3 displays dialog boxes.

## Working with the Methods Tables

The Methods table for an object lists the following:

- The method name
- Parameter(s)

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

Appeasements can be required or optional. Optional parameters are indicated in the table by square brackets ([]). See ['Working with Method Parameters' on page 36](#) for information on using parameters.

- Return value type(s)
- A description, if applicable

Method	Parameter Type	Returns	What it does
<b>executeAction</b> (eventID [, descriptor] [, displayDialogs])	number (long) <a href="#">ActionDescriptor</a> <a href="#">DialogModes</a>	<a href="#">ActionDescriptor</a>	Plays an ActionManager event.

## Working with Method Parameters

Optional parameters are surrounded by square brackets ([ ]). In the following Methods table sample, the parameters `descriptor` and `displayDialogs` are optional and the parameter `eventID` is not.

Therefore, if you use the `executeAction()` method for the object associated with the sample Methods table above, you *must* include an `eventID` value in the parentheses following the method name. The `eventID` value must be a number, as indicated by the `number (long)` in the table's Parameter Type column.

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

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

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

```
app.executeAction(4233,,error)
```

# ActionDescriptor

A record of key-value pairs for actions, such as those included on the Adobe Photoshop CS3 Actions menu.

**Note:** The `ActionDescriptor` class is part of the Action Manager functionality. For more details on the Action Manager, see the *Photoshop CS3 Scripting Guide*.

## Properties

Property	Value Type	What it is
<b>count</b>	number (long)	Read-only. The number of keys contained in the descriptor.
<b>typename</b>	string	Read-only. The class name of the referenced <code>actionDescriptor</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>clear</b> ( )			Clears the descriptor.
<b>erase</b> (key)	number (long)		Erases a key from the descriptor.
<b>fromStream</b> (value)	string		Creates a descriptor from a stream of bytes; for reading from disk.
<b>getBoolean</b> (key)	number (long)	boolean	Gets the value of a key of type boolean.
<b>getClass</b> (key)	number (long)	number (long)	Gets the value of a key of type class.
<b>getData</b> (key)	number (long)	string	Gets raw byte data as a string value.
<b>getDouble</b> (key)	number (long)	number (double)	Gets the value of a key of type double.
<b>getEnumerationType</b> (key)	number (long)	number (long)	Gets the enumeration type of a key.
<b>getEnumerationValue</b> (key)	number (long)	number (long)	Gets the enumeration value of a key.
<b>getInteger</b> (key)	number (long)	number (long)	Gets the value of a key of type integer.
<b>getKey</b> (index)	number (long)	number (long)	Gets the ID of the <i>N</i> th key, provided by index.
<b>getList</b> (key)	number (long)	<a href="#">ActionList</a>	Gets the value of a key of type list.

Method	Parameter Type	Returns	What it does (Continued)
<b>getObjectType</b> (key)	number (long)	number (long)	Gets the class ID of an object in a key of type object.
<b>getObjectValue</b> (key)	number (long)	<a href="#">ActionDescriptor</a>	Gets the value of a key of type object.
<b>getPath</b> (key)	number (long)	File	Gets the value of a key of type File.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>getReference</b> (key)	number (long)	<a href="#">ActionReference</a>	Gets the value of a key of type <a href="#">ActionReference</a> .
<b>getString</b> (key)	number (long)	string	Gets the value of a key of type string.
<b>getType</b> (key)	number (long)	<a href="#">DescValueType</a>	Gets the type of a key.
<b>getUnitDoubleType</b> (key)	number (long)	number (long)	Gets the unit type of a key of type UnitDouble.
<b>getUnitDoubleValue</b> (key)	number (long)	number (double)	Gets the value of a key of type UnitDouble.
<b>hasKey</b> (key)	number (long)	boolean	Checks whether the descriptor contains the provided key.
<b>isEqual</b> (otherDesc)	<a href="#">ActionDescriptor</a>	boolean	Determines whether the descriptor is the same as another descriptor.
<b>putBoolean</b> (key, value)	number (long) boolean		Sets the value for a key whose type is boolean.
<b>putClass</b> (key, value)	number (long) number (long)		Sets the value for a key whose type is class.
<b>putData</b> (key, value)	number (long) string		Puts raw byte data as a string value.
<b>putDouble</b> (key, value)	number (long) number (double)		Sets the value for a key whose type is double.
<b>putEnumerated</b> (key, enumType, value)	number (long) number (long) number (long)		Sets the enumeration type and value for a key.
<b>putInteger</b> (key, value)	number (long) number (long)		Sets the value for a key whose type is integer.

Method	Parameter Type	Returns	What it does (Continued)
<b>putList</b> (key, value)	number (long) <a href="#">ActionList</a>		Sets the value for a key whose type is an <code>ActionList</code> object.
<b>putObject</b> (key, classID, value)	number (long) number (long) <a href="#">ActionDescriptor</a>		Sets the value for a key whose type is an object, represented by an Action Descriptor.
<b>putPath</b> (key, value)	number (long) File		Sets the value for a key whose type is path.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<b>putReference</b> (key, value)	number (long) <a href="#">ActionReference</a>		Sets the value for a key whose type is an object reference.
<b>putString</b> (key, value)	number (long) string		Sets the value for a key whose type is string.
<b>putUnitDouble</b> (key, unitID, value)	number (long) number (long) number (double)		Sets the value for a key whose type is a unit value formatted as a double.
<b>toStream</b>		string	Gets the entire descriptor as a stream of bytes; for writing from disk.

# ActionList

The list of commands that comprise an Action (such as an Action created using the Actions palette in the Adobe Photoshop CS3 application).

**Note:** The `ActionList` object is part of the Action Manager functionality. For details on using the Action Manager, see the *Photoshop CS3 Scripting Guide*.

## Properties

Property	Value Type	What it is
<code>count</code>	number (long)	Read-only. The number of commands that comprise the action.
<code>typename</code>	string	Read-only. The class name of the referenced <code>ActionList</code> object.

## Methods

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

Method	Parameter Type	Returns	What it does
<code>clear</code> ( )			Clears the list.
<code>getBoolean</code> (index)	number (long)	boolean	Gets the value of a list item of type boolean.
<code>getClass</code> (index)	number (long)	number (long)	Gets the value of a list item of type class.
<code>getData</code> (index)	number (long)	string	Gets raw byte data as a string value.
<code>getDouble</code> (index)	number (long)	number (double)	Gets the value of a list item of type double.
<code>getEnumerationType</code> (index)	number (long)	number (long)	Gets the enumeration type of a list item.
<code>getEnumerationValue</code> (index)	number (long)	number (long)	Gets the enumeration value of a list item.
<code>getInteger</code> (index)	number (long)	number (long)	Gets the value of a list item of type integer.
<code>getList</code> (index)	number (long)	<a href="#">ActionList</a>	Gets the value of a list item of type list.
<code>getObjectType</code> (index)	number (long)	number (long)	Gets the class ID of a list item of type object.



Method	Parameter Type	Returns	What it does (Continued)
<b>getObjectValue</b> (index)	number (long)	<a href="#">ActionDescriptor</a>	Gets the value of a list item of type object.
<b>getPath</b> (index)	number (long)	File	Gets the value of a list item of type File.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>getReference</b> (index)	number (long)	<a href="#">ActionReference</a>	Gets the value of a list item of type <a href="#">ActionReference</a> .
<b>getString</b> (index)	number (long)	string	Gets the value of a list item of type string.
<b>getType</b> (index)	number (long)	<a href="#">DescValueType</a>	Gets the type of a list item.
<b>getUnitDoubleType</b> (index)	number (long)	number (long)	Gets the unit value type of a list item of type Double.
<b>getUnitDoubleValue</b> (index)	number (long)	number (double)	Gets the unit value of a list item of type double.
<b>putBoolean</b> (value)	boolean		Sets the value to either true or false.
<b>putClass</b> (value)	number (long)		Sets the class or data type.
<b>putData</b> (value)	string		Puts raw byte data as a string value.
<b>putDouble</b> (value)	number (double)		Sets the value type as a double.
<b>putEnumerated</b> (enumType, value)	number (long) number (long)		Sets the value type as an enumerated, or constant, value.
<b>putInteger</b> (value)	number (long)		Sets the value of a list item of type integer.
<b>putList</b> (value)	<a href="#">ActionList</a>		Sets the value of a list item of type list or array.
<b>putObject</b> (classID, value)	number (long) <a href="#">ActionDescriptor</a>		Sets the value of a list item of type object.
<b>putPath</b> (value)	File		Sets the value of a list item of type path.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>

Method	Parameter Type	Returns	What it does (Continued)
<b>putReference</b> (value)	<a href="#">ActionReference</a>		Sets the value of a list item whose type a reference to an object created in the script.
<b>putString</b> (value)	string		Sets the value of a list item of type string.
<b>putUnitDouble</b> (classID, value)	number (long) number (double)		Sets the value of a list item of type unit value represented as a double.

# ActionReference

Contains data describing a referenced Action.

**Note:** The `ActionReference` object is part of the Action Manager functionality. For details on using the Action Manager, see the *Photoshop CS3 Scripting Guide*.

## Properties

Property	Value type	What it does
<code>typename</code>	string	Read-only. The class name of the referenced Action object.

## Methods

Method	Parameter Type	Returns	What it does
<code>getContainer</code> ( )		<a href="#">ActionReference</a>	Gets a reference contained in this reference.  Container references provide additional pieces to the reference. This looks like another reference, but it is actually part of the same reference.
<code>getDesiredClass</code> ( )		number (long)	Gets a number representing the class of the object.
<code>getEnumeratedType</code> ( )		number (long)	Gets the enumeration type.
<code>getEnumeratedValue</code> ( )		number (long)	Gets the enumeration value.
<code>getForm</code> ( )		<a href="#">ReferenceFormType</a>	Gets the form of an <a href="#">ActionReference</a> .
<code>getIdentifier</code> ( )		number (long)	Gets the identifier value for a reference whose form is identifier.
<code>getIndex</code> ( )		number (long)	Gets the index value for a reference in a list or array.
<code>getName</code> ( )		string	Gets the name of a reference.
<code>getOffset</code> ( )		number (long)	Gets the offset of the object's index value.
<code>getProperty</code> ( )		number (long)	Gets the property ID value.
<code>putClass</code> (desiredClass)	number (long)		Puts a new class form and class type into the reference.

Method	Parameter Type	Returns	What it does (Continued)
<b>putEnumerated</b> (desiredClass, enumType, value)	number (long) number (long) number (long)		Puts an enumeration type and ID into a reference along with the desired class for the reference.
<b>putIdentifier</b> (desiredClass, value)	number (long) number (long)		Puts a new identifier and value into the reference..
<b>putIndex</b> (desiredClass, value)	number (long) number (long)		Puts a new index and value into the reference.
<b>putName</b> (desiredClass, value)	number (long) string		Puts a new name and value into the reference.
<b>putOffset</b> (desiredClass, value)	number (long) number (long)		Puts a new offset and value into the reference.
<b>putProperty</b> (desiredClass, value)	number (long) number (long)		Puts a new property and value into the reference.

# Application

The Adobe Adobe Photoshop CS3 application object, which contains all other Adobe Photoshop CS3 objects.

**Note:** Because you open JavaScripts through the application itself, you do not need to use the `Application` object as part of the containment hierarchy that describes an object.

However, if you choose to include the `Application` object in your code, you must use the pre-defined global object name `app`, rather than the class name `Application`, in a script, as in the following sample:

```
var docRef = app.documents.add(800, 600, 72, "docRef", NewDocumentMode.RGB)
```

The following sample uses the `Application` object incorrectly:

```
var docRef = Application.documents.add(800, 600, 72, "docRef",  
NewDocumentMode.RGB)
```

However, the most common way to add an element in your code is to omit references to the `Application` object altogether, as in the following sample:

```
var docRef = documents.add(800, 600, 72, "docRef", NewDocumentMode.RGB)
```

## Properties

Property	Value Type	What it is
<code>activeDocument</code>	<a href="#">Document</a>	Read-write. The frontmost document. (Setting this property is equivalent to clicking an open document in the Adobe Photoshop CS3 application to bring it to the front of the screen.)
<code>backgroundColor</code>	<a href="#">SolidColor</a>	Read-write. The color mode for the document's background color.
<code>colorSettings</code>	String	Read-write. The name of selected color setting's set.
<code>displayDialogs</code>	<a href="#">DialogModes</a>	Read-write. The dialog mode for the document, which indicates whether or not Adobe Photoshop CS3 displays dialogs when the script runs.
<code>documents</code>	<a href="#">Documents</a>	Read-only. The collection of open documents.
<code>fonts</code>	<a href="#">TextFonts</a>	Read-only. The fonts installed on this system.
<code>foregroundColor</code>	<a href="#">SolidColor</a>	Read-write. The default foreground color (used to paint, fill, and stroke selections).
<code>freeMemory</code>	number (double)	Read-only. The amount of unused memory available to Adobe Photoshop CS3.
<code>locale</code>	string	Read-only. The language location of the application.
<code>macintoshFileTypes</code>	array of strings	Read-only. A list of file image types Adobe Photoshop CS3 can open.
<code>measurementLog</code>	<a href="#">MeasurementLog</a>	The log of measurements taken.

Property	Value Type	What it is (Continued)
<b>name</b>	string	Read-only. The application's name.
<b>notifiers</b>	<a href="#">Notifiers</a>	Read-only. The collection of notifiers currently configured (in the Scripts Events Manager menu in the Adobe Photoshop CS3 application).
<b>notifiersEnabled</b>	boolean	Read-write. Indicates whether all notifiers are enabled or disabled.
<b>path</b>	File	Read-only. The full path to the location of the Adobe Photoshop CS3 application.
<b>playbackDisplayDialogs</b>	<a href="#">DialogModes</a>	Read-write. The dialog mode for playback mode, which indicates whether or not Adobe Photoshop CS3 displays dialogs in playback mode.
<b>playbackParameters</b>	<a href="#">ActionDescriptor</a>	Read-write. The playback options, which indicate the speed at which Adobe Photoshop CS3 plays actions.
<b>preferences</b>	<a href="#">Preferences</a>	Read-only. The application preference settings (equivalent to selecting <b>Edit &gt; Preferences</b> in the Adobe Photoshop CS3 application in Windows or Photoshop > Preferences in Mac OS).
<b>preferencesFolder</b>	File	Read-only. The full path to the Preferences folder.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>recentFiles</b>	array of File	Read-only. Files in the Recent Files list.
<b>scriptingBuildDate</b>	string	Read-only. The build date of the Scripting interface.
<b>scriptingVersion</b>	string	Read-only. The version of the Scripting interface.
<b>typename</b>	string	Read-only. The class name of the referenced app object.
<b>version</b>	string	Read-only. The version of Adobe Photoshop application you are running.
<b>windowsFileTypes</b>	array of strings	Read-only. A list of file image extensions Adobe Photoshop CS3 can open.

## Methods

Method	Parameter Type	Returns	What it does
<b>batch</b> (inputFiles, action, from [, options])	array of File string string <a href="#">BatchOptions</a>	string	Runs the batch automation routine ( <b>similar to the Batch command, or File &gt; Automate &gt; Batch in the Adobe Photoshop CS3 application</b> ).  <b>Note:</b> The <code>inputFiles</code> parameter specifies the source for the files to be manipulated by the Batch command.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>beep</b> ()			Causes a “beep” sound.
<b>bringToFront</b>			Makes Adobe Photoshop CS3 the active (front-most) application.
<b>charIDToTypeID</b> (charID)	string	number (long)	Converts from a four character code (character ID) to a runtime ID.
<b>doAction</b> (action, from)	string string		Plays an action from the Actions palette.
<b>eraseCustomOptions</b> (key )	string		Erases user object with ID value <code>key</code> from the Photoshop registry.
<b>executeAction</b> (eventID [, descriptor] [, displayDialogs])	number (long) <a href="#">ActionDescriptor</a> <a href="#">DialogModes</a>	<a href="#">ActionDescriptor</a>	Plays an ActionManager event.
<b>executeActionGet</b> (reference)	<a href="#">ActionReference</a>	<a href="#">ActionDescriptor</a>	Obtains an ActionDescriptor.

Method	Parameter Type	Returns	What it does (Continued)
<b>featureEnabled</b> (name )	string	boolean	Determines whether the feature specified by <code>name</code> is enabled.  The following features are supported as values for <code>name</code> : "photoshop/extended" "photoshop/standard" "photoshop/trial"
<b>getCustomOptions</b> (key )	string	<a href="#">ActionDescriptor</a>	Retrieves user objects in the Photoshop registry for the ID with value <code>key</code> .
<b>load</b> (document)	File		Loads the support document from the specified location.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>makeContactSheet</b> (inputFiles [, options])	array of File <a href="#">ContactSheetOptions</a>	string	Creates a contact sheet from the specified files.
<b>makePDFPresentation</b> (inputFiles, outputFiles [, options])	array of File File <a href="#">PresentationOptions</a>	string	Creates an Adobe PDF presentation file from the specified input files.  <b>Note:</b> The return string contains the path to the PDF file.
<b>makePhotoGallery</b> (inputFolder, outputFolder [, options])	File File <a href="#">GalleryOptions</a>	string	Creates a web photo gallery from the files in the specified input folder.
<b>makePhotomerge</b> (inputFiles)	array of File	string	Deprecated for Adobe Photoshop CS3. Use:  <pre>runphotomergeFromScript = true; \$.evalFile( app.Path + "Presets/Scripts/Photomerge.jsx"); photomerge.createPanorama( fileList, displayDialog );</pre> Merges multiple files into one; user interaction required.
<b>makePicturePackage</b> (inputFiles [, options])	array of File <a href="#">PicturePackageOptions</a>	string	Creates a picture package from the specified input files.



Method	Parameter Type	Returns	What it does (Continued)
<b>open</b> (document [, as] [, asSmartObject] )	File object (open options) boolean  <b>Note:</b> See individual file type open options, such as <a href="#">CameraRAWOpenOptions</a> or <a href="#">EPSOpenOptions</a> , etc.	<a href="#">Document</a>	Opens the specified document as the optionally specified file type. Optional parameter <code>asSmartObject</code> (default: <code>false</code> ) indicates whether to create a smart object around the opened document.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i> .  See the Application sample script for an example of using the <code>File</code> object in the <code>open</code> method.
<b>openDialog</b> ()		array of <code>File</code>	Uses the Photoshop open dialog box to select files.  Returns an array of <code>File</code> representing the files selected.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools</i> <i>Guide</i>
<b>purge</b> (target)	<a href="#">PurgeTarget</a>		Purges one or more caches.
<b>putCustomOptions</b> (key, customObject [, persistent] )	string <a href="#">ActionDescriptor</a> boolean		Saves user objects in the Photoshop registry.  <code>key</code> provides the unique ID for your user object.  <code>customObject</code> provides the object to save in the registry.  <code>persistent</code> indicates whether the object should persist once the script has finished.
<b>refresh</b> ()			Pauses the script while the application refreshes.
<b>stringIDToTypeID</b> (stringID)	string	number (long)	Converts from a string ID to a runtime ID.

Method	Parameter Type	Returns	What it does (Continued)
<b>typeIDToCharID</b> (typeID)	number (long)	string	Converts from a runtime ID to a character ID.
<b>typeIDToStringID</b> (typeID)	number (long)	string	Converts from a runtime ID to a string ID.

## First Sample Script

The following script invokes an alert box to display Properties important to an application such as version number, the path to the application, the amount of memory available, and the number of documents open.

When a user presses the OK button on the alert box, a second dialog opens, which asks users whether they would like the foreground and background colors set for the document presently open. If no document is open, the script opens a new document for the user.

The script (with no document open) produces a progression of three dialogs.

### Application.jsx

```
//Create a Welcome message
// Use the name and version properties of the application object to
// Append the application's name and version to the Welcome message
// use "\r" to insert a carriage return
// use the combination operator += to append info to the message
var message = "Welcome to " + app.name
message += " version " + app.version + "\r\r"

// find out where Adobe Photoshop CS3 is installed
// and add the path to the message
// add the optional parameter fsName to the path property
// to display the file system name in the most common format
message += "I'm installed in " + app.path.fsName + "\r\r"

// see how much memory Adobe Photoshop CS3 has to play with
message += "You have this much memory available for Adobe Photoshop CS3: " +
app.freeMemory + "\r\r"

// use the length property of the documents object to
// see how many documents are open
var documentsOpen = app.documents.length
message += "You currently have " + documentsOpen + " document(s) open.\r\r"

// display the message to the user
alert(message)

// answer will be true for a "Yes" answer and false for a "No" answer
var answer = confirm("Do you want me to set the foreground and background to my
favorite colors?")

// set the colors
if (answer) {
    // I don't have a favorite color. Why did I ask you may wonder?
    app.foregroundColor.rgb.red = Math.random() * 255
    app.foregroundColor.rgb.green = Math.random() * 255
    app.foregroundColor.rgb.blue = Math.random() * 255
}
```

```
app.backgroundColor.rgb.red = Math.random() * 255
app.backgroundColor.rgb.green = Math.random() * 255
app.backgroundColor.rgb.blue = Math.random() * 255
}

// Open a document
if (app.documents.length == 0) {

    // use the application's path and the offset to the samples folder
    var sampleDocToOpen = File(app.path + "/Samples/Fish.psd")

    // compose a message with the name of the file
    message = "Would you like me to open a sample for you? ("
    message += sampleDocToOpen.fsName
    message += ")"

    // ask the user another question
    answer = confirm(message)

    // open the document accordingly
    if (answer) {
        open(sampleDocToOpen)
    }
}
```

## Second Sample Script

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

### PDFPresentation.jsx

```
// use all the files in the Samples folder
var inputFolder = new Folder(app.path + "/Samples/")

// see if we have something interesting
if (inputFolder != null) {

    // get all the files found in this folder that are Adobe Photoshop CS3 (.psd
    format)
    var inputFiles = inputFolder.GetFiles("*.psd")

    // output to the desktop
    var outputFile = File("~/Desktop/JavaScriptPresentation.pdf")

    // there are defaults but I like to set the options myself
    var options = new PresentationOptions
    options.presentation = true
    options.view = true
    options.autoAdvance = true
    options.interval = 5
    options.loop = true
    options.transition = TransitionType.RANDOM

    // create the presentation
    makePDFPresentation(inputFiles, outputFile, options)

    alert("Presentation file saved to: " + outputFile.fsName)
}
```

**Note:** To run this code on non-English platforms, substitute the following path for the `outputFile` variable:

```
var outputFile = File("~/JavaScriptPresentation.pdf")
```

# ArtLayer

An object within a document that contains the visual elements of the image (equivalent to a layer in the Adobe Photoshop CS3 application).

**Note:** Most likely, you will use variables to refer to `ArtLayer` objects in your script. However, if you choose not to use a variable, be aware that, because the `ArtLayer` class is also a property of the [Document](#) object, you use the property name, `artLayer`, rather than the class name, `ArtLayer`, in your code.

The following example uses correct syntax to refer to an `ArtLayer` object by name and then assign its `allLocked` property value:

```
documents(0).artLayer("my layer").allLocked = true
```

The following example, which uses an upper case A in the object name, is incorrect:

```
documents(0).ArtLayer("my layer").allLocked = true
```

## Properties

Property	Value Type	What it is
<code>allLocked</code>	boolean	Read-write. Indicates whether to completely lock the layer's contents and settings.
<code>blendMode</code>	<a href="#">BlendMode</a>	Read-write. The layer's blending mode.
<code>bounds</code>	array (UnitValue)	Read-only. An array of coordinates that describes the bounding rectangle of the layer.  <b>Note:</b> For information about the UnitValue type, see the <i>JavaScript Tools Guide</i> .
<code>fillOpacity</code>	number (double)	Read-write. The interior opacity of the layer (between 0.0 and 100.0).
<code>grouped</code>	boolean	Read-write. Indicates this layer is grouped with the layer beneath it.
<code>isBackgroundLayer</code>	boolean	Read-write. Indicates whether the layer is a background layer or normal layer.  <b>Note:</b> A document can have only one background layer.
<code>kind</code>	<a href="#">LayerKind</a>	Read-write. Sets the layer's kind (such as 'text layer') for an empty layer.  <b>Note:</b> Valid only when the layer is empty and when <code>isBackgroundLayer</code> is <code>false</code> . See <a href="#">isBackgroundLayer</a> .  <b>Note:</b> 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 set <code>isBackgroundLayer</code> to <code>true</code> .

Property	Value Type	What it is (Continued)
<b>linkedLayers</b>	array of <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> objects	Read-only. The layers linked to this layer. <b>Note:</b> See <a href="#">link</a> .
<b>name</b>	string	Read-write. The layer's name.
<b>opacity</b>	number (double)	Read-write. The master opacity of the layer (0.0 - 100.0).
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The object's container.
<b>pixelsLocked</b>	boolean	Read-write. Indicates whether the pixels in the layer's image can be edited using the paintbrush tool.
<b>positionLocked</b>	boolean	Read-write. Indicates whether the pixels in the layer's image can be moved within the layer.
<b>textItem</b>	<a href="#">TextItem</a>	Read-only. The text item that is associated with the layer. <b>Note:</b> Valid only when <code>kind = LayerKind.TEXT</code> . See <a href="#">kind</a> .
<b>transparentPixelsLocked</b>	boolean	Read-write. Indicates whether editing is confined to the opaque portions of the layer.
<b>typename</b>	string	Read-only. The class name of the referenced <code>artLayer</code> object.
<b>visible</b>	boolean	Read-write. Indicates whether the layer is visible.

## Methods

Method	Parameter Type	Returns	What it does
<b>adjustBrightnessContrast</b> (brightness, contrast)	number (long) number (long)		Adjusts the brightness (-100 - 100) and contrast (-100 - 100).
<b>adjustColorBalance</b> ([shadows] [, midtones] [, highlights] [, preserveLuminosity])	array of integers array of integers array of integers boolean		Adjusts the color balance of the layer's component channels. For shadows, midtones, and highlights, the array must include three values (-100 - 100), which represent cyan or red, magenta or green, and yellow or blue, when the document mode is CMYK or RGB.  <b>Note:</b> See mode in the Properties table of the <a href="#">Document</a> object.

Method	Parameter Type	Returns	What it does (Continued)
<b>adjustCurves</b> (curveShape)	array of points (Array (Array(x, y)))		Adjusts the tonal range of the selected channel using up to fourteen points.
<b>adjustLevels</b> (inputRangeStart, inputRangeEnd, inputRangeGamma, outputRangeStart, outputRangeEnd)	number (long) number (long) number (double) number (long) number (long)		Adjusts the levels of the selected channels (inputRangeStart: 0 - 253; inputRangeEnd: (inputRangeStart + 2) - 255; inputRangeGamma: 0.10 - 9.99; outputRangeStart: 0 - 253; outputRangeEnd: (outputRangeStart + 2) - 255.
<b>applyAddNoise</b> (amount, distribution, monochromatic)	number (double) <a href="#">NoiseDistribution</a> boolean		Applies the Add Noise filter (amount: 0.1 - 400, as a percentage).
<b>applyAverage</b> ( )			Applies the Average filter.
<b>applyBlur</b> ( )			Applies the Blur filter.
<b>applyBlurMore</b> ( )			Applies the Blur More filter.
<b>applyClouds</b> ( )			Applies the Clouds filter.
<b>applyCustomFilter</b> (characteristics, scale, offset)	array of twenty-five numbers (long) number (long) number (long)		Applies a custom filter.  <b>Note:</b> Required parameter values define the filter. Refer to Adobe Photoshop CS3 Help for specific instructions.
<b>applyDeInterlace</b> (eliminateFields, createFields)	<a href="#">EliminateFields</a> <a href="#">CreateFields</a>		Applies the De-Interlace filter.
<b>applyDespeckle</b> ( )			Applies the Despeckle filter.
<b>applyDifferenceClouds</b> ( )			Applies the Difference Clouds filter.
<b>applyDiffuseGlow</b> (graininess, glowAmount, clearAmount)	number (long) number (long) number (long)		Applies the Diffuse Glow filter (graininess: 0 - 10; glowAmount: 0 - 20; clearAmount: 0 - 20).

Method	Parameter Type	Returns	What it does (Continued)
<b>applyDisplace</b> (horizontalScale, verticalScale, displacement, undefinedAreas, displacementMapFiles)	number (long) number (long) <a href="#">DisplacementMapType</a> <a href="#">UndefinedAreas</a> File		Applies the Displace filter using the specified horizontal and vertical scale (-999 - 999), mapping type, treatment of undistorted areas, and path to the distortion image map.
<b>applyDustAndScratches</b> (radius, threshold)	number (long) number (long)		Applies the Dust & Scratches filter (radius: 1 - 100; threshold: 0 - 255).
<b>applyGaussianBlur</b> (radius)	number (double)		Applies the Gaussian Blur filter within the specified radius (in pixels) (0.1 - 250.0).
<b>applyGlassEffect</b> (distortion, smoothness, scaling [, invert] [, texture] [, textureFile])	number (long) number (long) number (long) boolean <a href="#">TextureType</a> File		Applies the Glass filter (distortion: 0 - 20; smoothness: 1 - 15; scaling (in percent): 50 - 200).  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<b>applyHighPass</b> (radius)	number (double)		Applies the High Pass filter within the specified radius (in pixels) (0.1 - 250.0).



Method	Parameter Type	Returns	What it does (Continued)
<b>applyLensBlur</b> ([source] [, focalDistance] [, invertDepthMap] [, shape] [, radius] [, bladeCurvature] [, rotation] [, brightness] [, threshold] [, amount] [, distribution] [, monochromatic] )	<a href="#">DepthMapSource</a> number (long) boolean <a href="#">Geometry</a> number (long) number (long) number (long) number (long) number (long) <a href="#">NoiseDistribution</a> boolean		<p>Applies the Lens Blur filter.</p> <p><b>source:</b> the source for the depth map (default: <code>DepthMapSource.NONE</code>)</p> <p><b>focalDistance :</b> the blur focal distance for the depth map (default: 0).</p> <p><b>invertDepthMask:</b> whether the depth map is inverted (default: false).</p> <p><b>shape:</b> The shape of the iris (default: <code>Geometry.HEXAGON</code>)</p> <p><b>radius:</b> The radius of the iris (default: 15).</p> <p><b>bladeCurvature:</b> The blade curvature of the iris (default: 0).</p> <p><b>rotation:</b> The rotation of the iris (default: 0)</p> <p><b>brightness:</b> The brightness for the specular highlights (default: 0).</p> <p><b>threshold:</b> The threshold for the specular highlights (default: 0).</p> <p><b>amount:</b> The amount of noise (default: 0)</p> <p><b>distribution:</b> The distribution value for the noise (default: <code>NoiseDistribution.UNIFORM</code>).</p> <p><b>monochromatic:</b> Indicates whether the noise is monochromatic (default: false).</p>
<b>applyLensFlare</b> (brightness, flareCenter, lensType)	number (long) array (UnitValue) <a href="#">LensType</a>		<p>Applies the Lens Flare filter with the specified brightness (0 - 300, as a percentage), the x and y coordinates (unit value) of the flare center, and the lens type.</p> <p><b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i></p>

Method	Parameter Type	Returns	What it does (Continued)
<b>applyMaximum</b> (radius)	number (double)		Applies the Maximum filter within the specified radius (in pixels) (1 - 100).
<b>applyMedianNoise</b> (radius)	number (double)		Applies the Median Noise filter within the specified radius (in pixels) (1 - 100).
<b>applyMinimum</b> (radius)	number (double)		Applies the Minimum filter within the specified radius (in pixels) (1 - 100).
<b>applyMotionBlur</b> (angle, radius)	number (long) number (double)		Applies the Motion Blur filter (angle: -360 - 360; radius: 1 - 999).
<b>applyNTSC</b> ( )			Applies the NTSC colors filter.
<b>applyOceanRipple</b> (size, magnitude)	number (long) number (long)		Applies the Ocean Ripple filter in the specified size (1 - 15) and magnitude (0 - 20).
<b>applyOffset</b> (horizontal, vertical, undefinedAreas)	UnitValue UnitValue <a href="#">OffsetUndefinedAreas</a>		Moves the layer the specified amount horizontally and vertically (min/max amounts depend on layer size), leaving an undefined area at the layer's original location.  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i>
<b>applyPinch</b> (amount)	number (long)		Applies the Pinch filter in the specified amount (as a percentage) (-100 - 100).
<b>applyPolarCoordinates</b> (conversion)	<a href="#">PolarConversionType</a>		Applies the Polar Coordinates filter.
<b>applyRadialBlur</b> (amount, blurMethod, blurQuality)	number (long) <a href="#">RadialBlurMethod</a> <a href="#">RadialBlurQuality</a>		Applies the Radial Blur filter in the specified amount (1 - 100) using either a spin or zoom effect and the specified quality.
<b>applyRipple</b> (amount, size)	number (long) <a href="#">RippleSize</a>		Applies the Ripple filter in the specified amount (-999 to 999) throughout the image and in the specified size.
<b>applySharpen</b> ( )			Applies the Sharpen filter.

Method	Parameter Type	Returns	What it does (Continued)
<b>applySharpenEdges</b> ( )			Applies the Sharpen Edges filter.
<b>applySharpenMore</b> ( )			Applies the Sharpen More filter.
<b>applyShear</b> (curve, undefinedAreas)	array of points (Array (Array(x, y))) <a href="#">UndefinedAreas</a>		Applies the Shear filter (curve: 2 - 255 points).
<b>applySmartBlur</b> (radius, threshold, blurQuality, mode)	number (double) number (double) <a href="#">SmartBlurQuality</a> <a href="#">SmartBlurMode</a>		Applies the Smart Blur filter (radius: 0.1 - 100.0; threshold: 0.1 - 100.0).
<b>applySpherize</b> (amount, mode)	number (long) <a href="#">SpherizeMode</a>		Applies the Spherize filter in the specified amount (as percentage) (-100 - 100).
<b>applyStyle</b> (styleName)	string		Applies the specified style to the layer.  <b>Note:</b> You must use a style from the Styles list in the Layer Styles Palette.
<b>applyTextureFill</b> (textureFile)	File		Applies the Texture Fill filter.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<b>applyTwirl</b> (angle)	number (long)		Applies the Twirl filter at the specified angle (-999 - 999).
<b>applyUnsharpMask</b> (amount, radius, threshold)	number (double) number (double) number (long)		Applies the Unsharp Mask filter (amount: 1 - 500 as percent; radius: 0.1 - 250.00; threshold: 0 - 255).
<b>applyWave</b> (generatorNumber, minimumWavelength, maximumWavelength, minimumAmplitude, maximumAmplitude, horizontalScale, verticalScale, waveType, undefinedAreas, randomSeed)	number (long) number (long) number (long) number (long) number (long) number (long) number (long) <a href="#">WaveType</a> <a href="#">UndefinedAreas</a> number (long)		Applies the Wave filter (generatorNumber: <b>1 - 999</b> ; minimumWavelength: <b>1 - 998</b> ; maximumWavelength: <b>2 - minimumWavelength + 1</b> ; minimumAmplitude: <b>1 - 998</b> ; maximumAmplitude: <b>2 - minimumAmplitude + 1</b> ; horizontalScale: <b>1 - 100</b> , as a percentage; verticalScale: <b>1 - 100</b> , as a percentage).

Method	Parameter Type	Returns	What it does (Continued)
<b>applyZigZag</b> (amount, ridges, style)	number (long) number (long) <a href="#">ZigZagType</a>		Applies the Zigzag filter (amount: -100 - 100; ridges: 0 - 20).
<b>autoContrast</b> ( )			Adjusts the contrast of the selected channels automatically.
<b>autoLevels</b> ( )			Adjusts the levels of the selected channels using the auto levels option.
<b>clear</b> ( )			Cuts the layer without moving it to the clipboard.
<b>copy</b> ([merge])	boolean		Copies the layer to the clipboard. When the optional argument is set to <code>true</code> , a merged copy is performed (that is, all visible layers are copied to the clipboard).
<b>cut</b> ( )			Cuts the layer to the clipboard.
<b>desaturate</b> ( )			Converts a color image to a grayscale image in the current color mode by assigning equal values of each component color to each pixel.
<b>duplicate</b> ([relativeObject]  [, insertionLocation])	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> ) <a href="#">ElementPlacement</a>	<a href="#">ArtLayer</a> or <a href="#">LayerSet</a>	Creates a duplicate of the object on the screen.
<b>equalize</b> ( )			Redistributes the brightness values of pixels in an image to more evenly represent the entire range of brightness levels within the image.
<b>invert</b> ( )			Inverts the colors in the layer by converting the brightness value of each pixel in the channels to the inverse value on the 256-step color-values scale.
<b>link</b> (with)	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> )		Links the layer with the specified layer.

Method	Parameter Type	Returns	What it does (Continued)
<b>merge</b> ( )		<a href="#">ArtLayer</a>	Merges the layer down, removing the layer from the document; returns a reference to the art layer that this layer is merged into.
<b>mixChannels</b> (outputChannels [, monochrome])	array of array of numbers (double) boolean		<p>Modifies a targeted (output) color channel using a mix of the existing color channels in the image. (outputChannels = An array of channel specifications. For each component channel, specify a list of adjustment values (-200 - 200) followed by a 'constant' value (-200 - 200).)</p> <p><b>Note:</b> When monochrome = true, the maximum number of channel value specifications is 1.</p> <p><b>Note:</b> Valid only when docRef.mode = DocumentMode.RGB or docRef.mode = DocumentMode.CMYK.</p> <p><b>Note:</b> RGB arrays must include four doubles. CMYK arrays must include five doubles.</p>
<b>move</b> (relativeObject, insertionLocation)	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> ) <a href="#">ElementPlacement</a>		<p>Moves the layer relative to the object specified in parameters.</p> <p><b>Note:</b> For art layers, only the constant values ElementPlacement. PLACEBEFORE and ElementPlacement. PLACEAFTER are valid.</p> <p>For layer sets, only the constant values ElementPlacement. PLACEBEFORE and ElementPlacement.INS IDE are valid.</p>

Method	Parameter Type	Returns	What it does (Continued)
<b>photoFilter</b> ([fillColor] [, density] [, preserveLuminosity])	<a href="#">SolidColor</a> number (long) boolean		Adjust the layer's color balance and temperature as if a color filter had been applied (density: 1 - 100, as a percentage).
<b>posterize</b> (levels)	number (long)		Specifies the number of tonal levels (2 - 255) for each channel and then maps pixels to the closest matching level.
<b>rasterize</b> (target)	<a href="#">RasterizeType</a>		Converts the targeted contents in the layer into a flat, raster image.
<b>remove</b> ()			Deletes the object.
<b>resize</b> ([horizontal] [, vertical] [, anchor])	number (double) number (double) <a href="#">AnchorPosition</a>		Resizes the layer to the specified dimensions (as a percentage of its current size) and places it in the specified position.
<b>rotate</b> (angle [, anchor])	number (double) <a href="#">AnchorPosition</a>		Rotates rotates the layer around the specified anchor point (default: <code>AnchorPosition.MIDDLECENTER</code> ).
<b>selectiveColor</b> (selectionMethod [, reds] [, yellows] [, greens] [, cyans] [, blues] [, magentas] [, whites] [, neutrals] [, blacks])	<a href="#">AdjustmentReference</a> array of numbers (long) array of numbers (long) array of numbers (long) array of numbers (long) array of numbers (long) array of numbers (long) array of numbers (long) array of numbers (long) array of numbers (long)		Modifies the amount of a process color in a specified primary color without affecting the other primary colors.  <b>Note:</b> Each color array must have four components.
<b>shadowHighlight</b> ([shadowAmount] [, shadowWidth] [, shadowRadius] [, highlightAmount] [, highlightWidth] [, highlightRadius] [, colorCorrection] [, midtoneContrast] [, blackClip] [, whiteClip])	number (long) number (long) number (long) number (long) number (long) number (long) number (long) number (double) number (double)		Adjusts the range of tones in the image's shadows and highlights (shadowAmount: 0 - 100 as percent; shadowWidth: 0 - 100 as percent; shadowRadius: 0 - 2500 in pixels; highlightAmount: 0 - 100 as percent; highlightWidth: 0 - 100 as percent; highlightRadius: 0 - 2500 in pixels; colorCorrection: -100 - 100; midtoneContrast: -100 - 100; blackClip: 0.000 - 50.000; whiteClip: 0.000 - 50.000).

Method	Parameter Type	Returns	What it does (Continued)
<b>threshold</b> (level)	number (long)		Converts grayscale or color images to high-contrast, B/W images by converting pixels lighter than the specified threshold to white and pixels darker than the threshold to black (level: 1 - 255).
<b>translate</b> ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the layer the specified amount (in pixels) relative to its current position.  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i>
<b>unlink</b> ( )			Unlinks the layer.

## Sample Script

The following script creates art layers to display a duck and a sand dune in an overlying checkerboard pattern. A multi-layered collage displays.

### ArtLayer.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS3 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

//Close all the open documents
while (app.documents.length) {
    app.activeDocument.close()
}

// Create a new document to merge all the samples into
var mergedDoc = app.documents.add(1000, 1000, 72, "Merged Samples",
NewDocumentMode.RGB, DocumentFill.TRANSPARENT, 1)

// Use the path to the application and append the samples folder
var samplesFolder = Folder(app.path + "/Samples/")

//Get all the files in the folder
var fileList = samplesFolder.GetFiles()

// open each file
for (var i = 0; i < fileList.length; i++) {
    // The fileList is folders and files so open only files
    if (fileList[i] instanceof File) {
```

```
open(fileList[i])

// use the document name for the layer name in the merged document
var docName = app.activeDocument.name

// flatten the document so we get everything and then copy
app.activeDocument.flatten()
app.activeDocument.selection.selectAll()
app.activeDocument.selection.copy()

// don't save anything we did
app.activeDocument.close(SaveOptions.DONOTSAVECHANGES)

// make a random selection on the document to paste into
// by dividing the document up in 4 quadrants and pasting
// into one of them by selecting that area
var topLeftH = Math.floor(Math.random() * 2)
var topLeftV = Math.floor(Math.random() * 2)
var docH = app.activeDocument.width.value / 2
var docV = app.activeDocument.height.value / 2
var selRegion = Array(Array(topLeftH * docH, topLeftV * docV),
    Array(topLeftH * docH + docH, topLeftV * docV),
    Array(topLeftH * docH + docH, topLeftV * docV + docV),
    Array(topLeftH * docH, topLeftV * docV + docV),
    Array(topLeftH * docH, topLeftV * docV))
app.activeDocument.selection.select(selRegion)
app.activeDocument.paste()

// change the layer name and opacity
app.activeDocument.activeLayer.name = docName
app.activeDocument.activeLayer.fillOpacity = 50
    }
}

// sort the layers by name
for (var x = 0; x < app.activeDocument.layers.length; x++) {
    for (var y = 0; y < app.activeDocument.layers.length - 1 - x; y++) {
        // Compare in a non-case sensitive way
        var doc1 = app.activeDocument.layers[y].name
        var doc2 = app.activeDocument.layers[y + 1].name
        if (doc1.toUpperCase() > doc2.toUpperCase()) {
            app.activeDocument.layers[y].move(app.activeDocument.layers[y+1],
                ElementPlacement.PLACEAFTER)
        }
    }
}

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs
```



## ArtLayers

The collection of `artLayer` objects in the document.

**Note:** Because the `ArtLayers` class is a property of the [Document](#) object, you use the property name, `artLayers`, rather than the class name, `ArtLayers`, in your code. For example:

```
var layerRef = docRef.artLayers.add()
```

The following sample uses the `ArtLayers` object incorrectly:

```
var layerRef = docRef.ArtLayers.add()
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>artLayers</code> collection.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>artLayers</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">ArtLayer</a>	Gets an element from the <code>artLayers</code> collection.
<b>add</b> ( )		<a href="#">ArtLayer</a>	Creates a new <code>artLayer</code> in the document.
<b>getByName</b> (name)	string	<a href="#">ArtLayer</a>	Get the first element in the <code>artLayers</code> collection with the provided name.
<b>removeAll</b> ( )		Nothing	Removes all elements from the <code>artLayers</code> collection.

## BatchOptions

Options to specify when running a Batch command.

**Note:** You specify the batch source folder as the `inputFiles` parameter of the `batch()` method, which is a method of the `Application` class. See [‘batch’ on page 47](#). JavaScript supports only folders as sources for batch commands.

### Properties

Property	Value type	What it is
<code>destination</code>	<a href="#">BatchDestinationType</a>	Read-write. The type of destination for the processed files (default: <code>BatchDestinationType.NODESTINATION</code> ).
<code>destinationFolder</code>	File	Read-write. The folder location for the processed files.  <b>Note:</b> Valid only when <code>destination = BatchDestinationType.FOLDER</code> . See <a href="#">destination</a> .  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<code>errorFile</code>	File	Read-write. The file in which to log errors encountered.  <b>Note:</b> To display errors on the screen (and stop batch processing when errors occur) leave blank.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<code>fileNaming</code>	array of <a href="#">FileNamingType</a> options	Read-write. A list of file naming options (maximum: 6).  <b>Note:</b> Valid only when <code>destination = BatchDestinationType.FOLDER</code> . See <a href="#">destination</a> .
<code>macintoshCompatible</code>	boolean	Read-write. Indicates whether to make the final file names Macintosh compatible (default: <code>true</code> ).  <b>Note:</b> Valid only when <code>destination = BatchDestinationType.FOLDER</code> . See <a href="#">destination</a> .
<code>overrideOpen</code>	boolean	Read-write. Indicates whether to override action open commands (default: <code>false</code> ).

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

# BitmapConversionOptions

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

**Note:** Convert color images to grayscale before converting the image to bitmap mode. See [‘desaturate’ on page 60](#) (in the Methods table of the `ArtLayer` object).

## Properties

Property	Value Type	What it is
<b>angle</b>	number (double)	Read-write. The angle (in degrees) at which to orient individual dots (-180 - 180). See <a href="#">shape</a> .  <b>Note:</b> Valid only when <code>method = BitmapConversionType.HALFTONESCREEN</code> . See <a href="#">method</a> .
<b>frequency</b>	number (double)	Read-write. The number of printer dots (per inch) to use (1.0 - 999.99).  <b>Note:</b> Valid only when <code>method = BitmapConversionType.HALFTONESCREEN</code> . See <a href="#">method</a> .
<b>method</b>	<a href="#">BitmapConversionType</a>	Read-write. The conversion method to use (default: <code>BitmapConversionType.DIFFUSIONDITHER</code> ).
<b>patternName</b>	string	Read-write. The name of the pattern to use.  For information about pre-installed valid patterns, see Adobe Photoshop CS3 Help on the bitmap conversion command, or view the options available in the Custom Color drop down box after choosing the bitmap conversion command.  <b>Note:</b> Valid only when <code>method = BitmapConversionType.CUSTOMPATTERN</code> . See <a href="#">method</a> .
<b>resolution</b>	number (double)	Read-write. The output resolution in pixels per inch (default: 72 . 0).
<b>shape</b>	<a href="#">BitmapHalfToneType</a>	Read-write. The dot shape to use.  <b>Note:</b> Valid only when <code>method = BitmapConversionType.HALFTONESCREEN</code> . See <a href="#">method</a> .
<b>typename</b>	string	Read-only. The class name of the referenced <code>bitmapConversionOptions</code> object.

# BMPSaveOptions

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

## Properties

Property	Value Type	What it is
<code>alphaChannels</code>	boolean	Read-write. Indicates whether to save the alpha channels.
<code>depth</code>	<a href="#">BMPDepthType</a>	Read-write. The number of bits per channel.
<code>flipRowOrder</code>	boolean	Read-write. Indicates whether to write the image from top to bottom (default: <code>false</code> ).  <b>Note:</b> Available only when <code>osType</code> = <code>OperatingSystem.WINDOWS</code> . See <a href="#">osType</a> .
<code>osType</code>	<a href="#">OperatingSystem</a>	Read-write. The target OS. (default: <code>OperatingSystem.WINDOWS</code> ).
<code>rleCompression</code>	boolean	Read-write. Indicates whether to use RLE compression.  <b>Note:</b> Available only when <code>osType</code> = <code>OperatingSystem.WINDOWS</code> . See <a href="#">osType</a> .
<code>typename</code>	string	Read-only. The class name of the referenced <code>BMPSaveOptions</code> object.

# CameraRAWOpenOptions

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

## Properties

Property	Value type	What it is
<b>bitsPerChannel</b>	<a href="#">BitsPerChannelType</a>	Read-write. The number of bits per channel.
<b>blueHue</b>	number (long)	Read-write. The blue hue of the shot (-100 - 100).
<b>blueSaturation</b>	number (long)	Read-write. The blue saturation of the shot (-100 - 100).
<b>brightness</b>	number (long)	Read-write. The brightness of the shot (0 - 150).
<b>chromaticAberrationBY</b>	number (long)	Read-write. The chromatic aberration B/Y of the shot (-100 - 100).
<b>chromaticAberrationRC</b>	number (long)	Read-write. The chromatic aberration R/C of the shot (-100 - 100).
<b>colorNoiseReduction</b>	number (long)	Read-write. The color noise reduction of the shot (0 - 100).
<b>colorSpace</b>	<a href="#">ColorSpaceType</a>	Read-write. The colorspace for the image.
<b>contrast</b>	number (long)	Read-write. The contrast of the shot (-50 - 100).
<b>exposure</b>	number (double)	Read-write. The exposure of the shot (4.0 - 4.0).
<b>greenHue</b>	number (long)	Read-write. The green hue of the shot (-100 - 100).
<b>greenSaturation</b>	number (long)	Read-write. The green saturation of the shot (-100 - 100).
<b>luminanceSmoothing</b>	number (long)	Read-write. The luminance smoothing of the shot (0 - 100).
<b>redHue</b>	number (long)	Read-write. The red hue of the shot (-100 - 100).
<b>redSaturation</b>	number (long)	Read-write. The red saturation of the shot (-100 - 100).
<b>resolution</b>	number (double)	Read-write. The resolution of the document in pixels per inch (1 - 999).
<b>saturation</b>	number (long)	Read-write. The saturation of the shot (-100 - 100).
<b>settings</b>	<a href="#">CameraRAWSettingsType</a>	Read-write. The global settings for all Camera RAW options. Default: <code>CameraRAWSettingsType.CAMERA</code> .
<b>shadows</b>	number (long)	Read-write. The shadows of the shot (0 - 100).

Property	Value type	What it is (Continued)
<b>shadowTint</b>	number (long)	Read-write. The shadow tint of the shot (-100 - 100).
<b>sharpness</b>	number (long)	Read-write. The sharpness of the shot (0 - 100).
<b>size</b>	<a href="#">CameraRAWSize</a>	Read-write. The size of the new document.
<b>temperature</b>	number (long)	Read-write. The temperature of the shot (2000 - 50000).
<b>tint</b>	number (long)	Read-write. The tint of the shot (-150 - 150).
<b>typename</b>	string	Read-only. The class name of the referenced <code>cameraRAWOpenOptions</code> object.
<b>vignettingAmount</b>	number (long)	Read-write. The vignetting amount of the shot (-100 - 100).
<b>vignettingMidpoint</b>	number (long)	Read-write. The vignetting mid point of the shot (-100 - 100).
<b>whiteBalance</b>	<a href="#">WhiteBalanceType</a>	Read-write. The white balance options for the image.

# Channel

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

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

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

**Note:** Most likely, you will use variables to refer to `Channel` objects in your script. However, if you choose not to use a variable, be aware that, because the `Channel` class is a property of the [Document](#) object, you use the property name, `channel`, rather than the class name, `Channel`, in your code.

The following example uses correct syntax to refer to a `Channel` object by name and then assign its `opacity` property value:

```
documents(0).channel("my channelr").opacity = 22
```

The following example, which uses an upper case C in the object name, is incorrect:

```
documents(0).Channel("my channelr").opacity = 22
```

## Properties

Property	Value Type	What it is
<b>color</b>	<a href="#">SolidColor</a>	Read-write. The color of the channel.  <b>Note:</b> Not valid when <code>type = ChannelType.COMPONENT</code> .
<b>histogram</b>	array of 256 numbers (long)	Read-only. A histogram of the color of the channel.  <b>Note:</b> Not valid when <code>type = ChannelType.COMPONENT</code> . For component channel histogram values, use the <a href="#">histogram</a> property of the <a href="#">Document</a> object instead.
<b>kind</b>	<a href="#">ChannelType</a>	Read-write. The channel type.
<b>name</b>	string	Read-write. The channel's name.
<b>opacity</b>	number (double)	Read-write. The opacity to use for alpha channels or the solidity to use for spot channels (0 - 100).  <b>Note:</b> Valid only when <code>type = ChannelType.MASKEDAREA</code> or <code>type = ChannelType.SELECTEDAREA</code> .
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The object's container.



Property	Value Type	What it is (Continued)
<b>typename</b>	string	Read-only. The class name of the referenced <code>channel</code> object.
<b>visible</b>	boolean	Read-write. Indicates whether the channel is visible.

## Methods

Method	Parameter Type	Returns	What it does
<b>duplicate</b> ([targetDocument])	<a href="#">Document</a>	<a href="#">Channel</a>	Duplicates the channel.
<b>merge</b> ( )			Merges a spot channel into the component channels.
<b>remove</b> ( )			Deletes the channel.

# Channels

The collection of `channel` objects in the document. See [Channel](#).

**Note:** Because the `Channels` class is also a property of the [Document](#) object, you use the property name, `channels`, rather than the class name, `Channels`, in your code. For example:

```
var channelRef = docRef.channels.add()
```

The following sample uses the `Channels` object incorrectly:

```
var channelRef = docRef.Channels.add()
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>channels</code> collection.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The <code>channels</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>channels</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">Channel</a>	Gets an element from the <code>channels</code> collection.
<b>add</b> ( )		<a href="#">Channel</a>	Creates a new <code>channel</code> object.
<b>getByName</b> (name)	string	<a href="#">Channel</a>	Get the first element in the <code>channels</code> collection with the provided name.
<b>removeAll</b> ( )			Removes all <code>channel</code> objects from the <code>channels</code> collection.

## Sample Script

The following script opens a file if one is not already open, and then writes a histogram report (`histogram.log`) for the channels in the active document.

**Note:** This script contains a switch construction that uses a `break` statement. The `break` statement requires an ending semicolon (;), as in the following sample:

```
break;
```

### Histogram.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs
```

```
// Set Adobe Photoshop CS3 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

// if there are no documents open then try to open a sample file
if (app.documents.length == 0) {
    open(File(app.path + "/Samples/Fish.psd"))
}

// get a reference to the working document
var docRef = app.activeDocument

// create the output file
// first figure out which kind of line feeds we need
if ($.os.search(/windows/i) != -1) {
    fileLineFeed = "windows"
} else {
    fileLineFeed = "macintosh"
}

// create the output file accordingly
fileOut = new File("~/Desktop/Histogram.log")
fileOut.lineFeed = fileLineFeed
fileOut.open("w", "TEXT", "????")

// write out a header
fileOut.write("Histogram report for " + docRef.name)

// find out how many pixels I have
var totalCount = docRef.width.value * docRef.height.value

// more info to the out file
fileOut.write(" with a total pixel count of " + totalCount + "\n")

// channel indexer
var channelIndex = 0

// remember which channels are currently active
var activeChannels = app.activeDocument.activeChannels

// document histogram only works in these modes
if (docRef.mode == DocumentMode.RGB ||
    docRef.mode == DocumentMode.INDEXEDCOLOR ||
    docRef.mode == DocumentMode.CMYK) {

    // activate the main channels so we can get the documents histogram
    TurnOnDocumentHistogramChannels(docRef)

    // Output the documents histogram
    OutputHistogram(docRef.histogram, "Luminosity", fileOut)
}

// local reference to work from
var myChannels = docRef.channels

// loop through each channel and output the histogram
for (var channelIndex = 0; channelIndex < myChannels.length; channelIndex++) {
```

```
// the channel has to be visible to get a histogram
myChannels[channelIndex].visible= true

// turn off all the other channels
for (var secondaryIndex = 0; secondaryIndex < myChannels.length;
    secondaryIndex++) {
    if (channelIndex != secondaryIndex) {
        myChannels[secondaryIndex].visible= false
    }
}

// Use the function to dump the histogram
OutputHistogram(myChannels[channelIndex].histogram,
    myChannels[channelIndex].name, fileOut)
}

// close down the output file
fileOut.close()
alert("Histogram file saved to: " + fileOut.fsName)

// reset the active channels
docRef.activeChannels = activeChannels

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs

// Utility function that takes a histogram and name
// and dumps to the output file
function OutputHistogram(inHistogram, inHistogramName, inOutFile) {

    // find ouch which count has the largest number
    // I scale everything to this number for the output
    var largestCount = 0

    // a simple indexer I can reuse
    var histogramIndex = 0

    // see how many samples we have total
    var histogramCount = 0

    // search through all and find the largest single item
    for (histogramIndex = 0; histogramIndex < inHistogram.length;
        histogramIndex++) {
        histogramCount += inHistogram[histogramIndex]
        if (inHistogram[histogramIndex] > largestCount)
            largestCount = inHistogram[histogramIndex]
    }

    // These should match
    if (histogramCount != totalCount) {
        alert("Something bad is happening!")
    }

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

    // output this data to the file
```

```
inOutFile.write("One X = " + pixelsPerX + " pixels.\n")

// output the name of this histogram
inOutFile.write(inHistogramName + "\n")

// loop through all the items and output in the following format
// 001
// 002
for (histogramIndex = 0; histogramIndex < inHistogram.length;
    histogramIndex++) {

    // I need an extra "0" for this line item to keep everything in line
    if (histogramIndex < 10)
        inOutFile.write("0")

    // I need an extra "0" for this line item to keep everything in line
    if (histogramIndex < 100)
        inOutFile.write("0")

    // output the index to file
    inOutFile.write(histogramIndex)

    // some spacing to make it look nice
    inOutFile.write(" ")

    // figure out how many X's I need
    var outputX = inHistogram[histogramIndex] / largestCount * 100

    // output the X's
    for (var a = 0; a < outputX; a++)
        inOutFile.write("X")

    inOutFile.write("\n")
}

inOutFile.write("\n")
}

// Function to active all the channels according to the documents mode
// Takes a document reference for input
function TurnOnDocumentHistogramChannels(inDocument) {

    // see how many channels we need to activate
    var visibleChannelCount = 0

    // based on the mode of the document
    switch (inDocument.mode) {

        case DocumentMode.BITMAP:
        case DocumentMode.GRAYSCALE:
        case DocumentMode.INDEXEDCOLOR:
            visibleChannelCount = 1
            break;

        case DocumentMode.DUOTONE:
            visibleChannelCount = 2
            break;

        case DocumentMode.RGB:
        case DocumentMode.LAB:
```

```
        visibleChannelCount = 3
        break;

    case DocumentMode.CMYK:
        visibleChannelCount = 4
        break;

    case DocumentMode.DUOTONE:
        visibleChannelCount = 4
        break;

    case DocumentMode.MULTICHANNEL:
    default:
        visibleChannelCount = inDocument.channels.length + 1
        break;
}

// now get the channels to activate into a local array
var aChannelArray = new Array()

// index for the active channels array
var aChannelIndex = 0

for(var channelIndex = 0; channelIndex < inDocument.channels.length;
    channelIndex++) {
    if (channelIndex < visibleChannelCount) {
        aChannelArray[aChannelIndex++] = inDocument.channels[channelIndex]
    }
}

// now activate them
inDocument.activeChannels = aChannelArray
}
```

# CMYKColor

The definition of a CMYK color.

## Properties

Property	Value Type	What it is
<b>black</b>	number (double)	Read-write. The black color value (as percent) (0.0 - 100.0).
<b>cyan</b>	number (double)	Read-write. The cyan color value (as percent) (0.0 - 100.0).
<b>magenta</b>	number (double)	Read-write. The magenta color value (as percent) (0.0 - 100.0).
<b>typename</b>	string	Read-only. The class name of the referenced <code>CMYKColor</code> object.
<b>yellow</b>	number (double)	Read-write. The yellow color value (as percent) (0.0 - 100.0).

## ColorSampler

A color sampler for the document.

**Note:** For additional information about color samplers, see Adobe Photoshop CS3 help on the Color SamplerTool.

### Properties

Property	Value type	What it is
<code>color</code>	<a href="#">SolidColor</a>	Read-only. The color of the color sampler.
<code>position</code>	array of <code>UnitValue</code>	Read-only. The position of the color sampler in the document. The array (x,y) represents the horizontal and vertical location of the count item.  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<code>parent</code>	object ( <a href="#">Document</a> )	Read-only. The <code>ColorSampler</code> object's container.
<code>typename</code>	string	Read-only. The class name of the referenced <code>ColorSampler</code> object.

### Methods

Method	Parameter type	Returns	What it does
<code>move</code> ( <code>position</code> )	array of <code>UnitValue</code>		Moves the color sampler to a new location in the document.  The <code>position</code> parameter (x,y) represents the new horizontal and vertical locations, respectively, of the moved color sampler.  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<code>remove</code> ( )			Deletes the <code>ColorSampler</code> object.



# ColorSamplers

The collection of `ColorSampler` objects in the document. See [ColorSampler](#).

**Note:** Because the `ColorSampler` class is also a property of the [Document](#) object, you use the property name, `colorSamplers`, rather than the class name, `ColorSamplers`, in your code. For example:

```
docRef.colorSamplers.removeAll()
```

The following sample uses the `ColorSamplers` object incorrectly:

```
docRef.ColorSamplers.removeAll()
```

## Properties

Property	Value Type	What it is
<code>length</code>	number (long)	Read-only. The number of elements in the <code>ColorSamplers</code> collection.
<code>parent</code>	object ( <a href="#">Document</a> )	Read-only. The <code>ColorSamplers</code> object's container.
<code>typename</code>	string	Read-only. The class name of the referenced <code>ColorSamplers</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<code>index</code> (itemKey)	number	<a href="#">ColorSampler</a>	Gets an element from the <code>ColorSamplers</code> collection.
<code>add</code> (position)	array of <code>UnitValue</code>	<a href="#">ColorSampler</a>	Creates a new <code>ColorSampler</code> object. The <code>position</code> parameter (x,y) represents the new horizontal and vertical locations, respectively, of the moved color sampler.  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<code>getByName</code> (name)	string	<a href="#">ColorSampler</a>	Get the first element in the <code>ColorSamplers</code> collection with the provided name.
<code>removeAll</code> ( )			Removes all <code>ColorSampler</code> objects from the <code>ColorSamplers</code> collection.

# ContactSheetOptions

Options that can be specified for a contact sheet.

## Properties

Property	Value Type	What it is
<b>acrossFirst</b>	boolean	Read-write. Indicates whether to place the images horizontally (left to right, then top to bottom) first (default: <code>true</code> ).
<b>bestFit</b>	boolean	Read-write. Indicates whether to rotate images for the best fit (default: <code>false</code> ).
<b>caption</b>	boolean	Read-write. Indicates whether to use the filename as a caption for the image (default: <code>true</code> ).
<b>columnCount</b>	number (long)	Read-write. The number of columns to include (1 - 100; default: 5).
<b>flatten</b>	boolean	Read-write. Indicates whether to flatten all layers in the final document (default: <code>true</code> ).
<b>font</b>	<a href="#">GalleryFontType</a>	Read-write. The font used for the caption (default: <code>GalleryFontType.ARIAL</code> ).
<b>fontSize</b>	number (long)	Read-write. The font size to use for the caption (default: 12).
<b>height</b>	number (long)	Read-write. The height (in pixels) of the resulting document (100 - 2900; default: 720).
<b>horizontal</b>	number (long)	Read-write. The horizontal spacing (in pixels) between images (0 - 29000; default: 1).
<b>mode</b>	<a href="#">NewDocumentMode</a>	Read-write. The document color mode (default: <code>NewDocumentMode.RGB</code> ).
<b>resolution</b>	number (double)	Read-write. The resolution of the document in pixels per inch (35 - 1200; default: 72.0).
<b>rowCount</b>	number (long)	Read-write. The number of rows to use (1 - 100; default: 6).
<b>typename</b>	string	Read-only. The class name of the referenced <code>contactSheetOptions</code> object.
<b>useAutoSpacing</b>	boolean	Read-write. Indicates whether to auto space the images (default: <code>true</code> ).

Property	Value Type	What it is (Continued)
<b>vertical</b>	number (long)	Read-write. The vertical spacing (in pixels) between images (0 - 29000; default: 1).  <b>Note:</b> Valid only when <code>useAutoSpacing = false</code> .
<b>width</b>	number (long)	Read-write. The width (in pixels) of the resulting document (100 - 2900; default: 576).

## CountItem

A counted item in the document. Also see the method [autoCount](#), defined on [Document](#).

**Note:** `CountItems` is available in the Extended Version only.

For additional information about count items, see Adobe Photoshop CS3 help on the Count Tool.

### Properties

Property	Value type	What it is
<code>position</code>	array of <code>UnitValue</code>	Read-only. The position of the count item in the document.  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<code>parent</code>	object ( <a href="#">Document</a> )	Read-only. The <code>CountItem</code> object's container.
<code>typename</code>	string	Read-only. The class name of the referenced <code>CountItem</code> object.

### Methods

Method	Parameter type	Returns	What it does
<code>remove</code> ( )			Deletes the <code>CountItem</code> object.

## CountItems

The collection of `CountItem` objects in the document. See [CountItem](#).

**Note:** `CountItem` is available in the Extended Version only.

Because the `CountItems` class is also a property of the [Document](#) object, you use the property name, `countItems`, rather than the class name, `CountItems`, in your code. For example:

```
docRef.countItems.removeAll()
```

The following sample uses the `CountItems` object incorrectly:

```
docRef.CountItems.removeAll()
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>CountItems</code> collection.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The <code>CountItems</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>CountItems</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">CountItem</a>	Gets an element from the <code>CountItems</code> collection.
<b>add</b> (position)	array of <code>UnitValue</code>	<a href="#">CountItem</a>	Creates a new <code>CountItem</code> object. Parameter <code>position (x,y)</code> represents the horizontal and vertical positions, respectively, of the <code>CountItem</code> object.  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<b>getByName</b> (name)	string	<a href="#">CountItem</a>	Get the first element in the <code>CountItems</code> collection with the provided name.
<b>removeAll</b> ( )			Removes all <code>CountItem</code> objects from the <code>CountItems</code> collection.

## DCS1\_SaveOptions

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

### Properties

Property	Value Type	What it is
<b>dcs</b>	<a href="#">DCSType</a>	Read-write. (default: <code>DCSType.COLORCOMPOSITE</code> ).
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document
<b>encoding</b>	<a href="#">SaveEncoding</a>	Read-write. The type of encoding to use for document (default: <code>SaveEncoding.BINARY</code> ).
<b>halftoneScreen</b>	boolean	Read-write. Indicates whether to include halftone screen (default: <code>false</code> ).
<b>interpolation</b>	boolean	Read-write. Indicates whether to use image interpolation (default: <code>false</code> )
<b>preview</b>	<a href="#">Preview</a>	Read-write. The type of preview (default: <code>Preview.MACOSEIGHTBIT</code> ).
<b>transferFunction</b>	boolean	Read-write. Indicates whether to include the Transfer functions to compensate for dot gain between the image and film (default: <code>false</code> ).
<b>typename</b>	string	Read-only. The class name of the referenced <code>DCS1_SaveOptions</code> object.
<b>vectorData</b>	boolean	Read-write. Indicates whether to include vector data.  <b>Note:</b> Valid only if the document includes vector data (un-rasterized text).

## DCS2\_SaveOptions

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

### Properties

Property	Value Type	What it is
<code>dcs</code>	<a href="#">DCSType</a>	Read-write. The type of composite file to create (default: <code>DCSType.NOCOMPOSITE</code> ).
<code>embedColorProfile</code>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<code>encoding</code>	<a href="#">SaveEncoding</a>	Read-write. The type of encoding to use (default: <code>SaveEncoding.BINARY</code> ).
<code>halftoneScreen</code>	boolean	Read-write. Indicates whether to include the halftone screen (default: <code>false</code> ).
<code>interpolation</code>	boolean	Read-write. Indicates whether to use image interpolation (default: <code>false</code> ).
<code>multiFileDCS</code>	boolean	Read-write. Indicates whether to save color channels as multiple files or a single file (default: <code>false</code> ).
<code>preview</code>	<a href="#">Preview</a>	Read-write. The preview type (default: <code>Preview.MACOSEIGHTBIT</code> ).
<code>spotColors</code>	boolean	Read-write. Indicates whether to save spot colors.
<code>transferFunction</code>	boolean	Read-write. Indicates whether to include the Transfer functions to compensate for dot gain between the image and film (default: <code>false</code> ).
<code>typename</code>	string	Read-only. The class name of the referenced <code>DCS2_SaveOptions</code> object.
<code>vectorData</code>	boolean	Read-write. Indicates whether to include vector data.  <b>Note:</b> Valid only if the document includes vector data (un-rasterized text).

## DICOMOpenOptions

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

**Note:** `DICOMOpenOptions` is available in the Extended Version only.

### Properties

Property	Value Type	What it is
<b>anonymize</b>	boolean	Read-write. Indicates whether to make the patient information anonymous.
<b>columns</b>	number (long)	Read-write. Number of columns in n-up configuration.
<b>reverse</b>	boolean	Read-write. Indicates whether to reverse (invert) the image.
<b>rows</b>	number (long)	Read-write. The number of rows in n-up configuration.
<b>showOverlays</b>	boolean	Read-write. Indicates whether to show overlays.
<b>typename</b>	string	Read-only. The class name of the referenced <code>DICOMOpenOptions</code> object.
<b>windowLevel</b>	number (long)	Read-write. The contrast of the image in Hounsfield units.
<b>windowWidth</b>	number (long)	Read-write. The brightness of the image in Hounsfield units.



# Document

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

**Note:** In Adobe Photoshop CS3, a document can also be referred to as an image or a canvas.

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

**Note:** Most likely, you will use variables to refer to `Document` objects in your script. However, if you choose not to use a variable, be aware that, because the `Document` class is a property of the [Application](#) object, you use the property name, `document`, rather than the class name, `Document`, in your code.

The following example uses correct syntax to refer to a `Document` object by name and then assign its `colorProfileType` property value:

```
document("my document").colorProfileType = ColorProfile.CUSTOM
```

The following example, which uses an upper case *D* in the object name, is incorrect:

```
Document("my document").colorProfileType = ColorProfile.CUSTOM
```

## Properties

Property	Value Type	What it is
<code>activeChannels</code>	array of <a href="#">Channel</a> objects	Read-write. The selected channels.
<code>activeHistoryBrushSource</code>	<a href="#">HistoryState</a>	Read-write. The history state to use with the history brush.
<code>activeHistoryState</code>	<a href="#">HistoryState</a>	Read-write. The selected <code>HistoryState</code> object.
<code>activeLayer</code>	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> )	Read-write. The selected layer.
<code>artLayers</code>	<a href="#">ArtLayers</a>	Read-only. The <code>artLayers</code> collection.
<code>backgroundLayer</code>	<a href="#">ArtLayer</a>	Read-only. The background layer of the document.
<code>bitsPerChannel</code>	<a href="#">BitsPerChannelType</a>	Read-write. The number of bits per channel.
<code>channels</code>	<a href="#">Channels</a>	Read-only. The <code>channels</code> collection.
<code>colorProfileName</code>	string	Read-write. The name of the color profile.  <b>Note:</b> Valid only when <code>colorProfileType = ColorProfile.CUSTOM</code> or <code>colorProfileType = ColorProfile.WORKING</code> . See <a href="#">colorProfileType</a> .
<code>colorProfileType</code>	<a href="#">ColorProfile</a>	Read-write. The type of color model that defines the document's working space.

Property	Value Type	What it is (Continued)
<code>colorSamplers</code>	<a href="#">ColorSamplers</a>	Read-only. The current color samplers associated with this document.
<code>componentChannels</code>	array of <a href="#">Channel</a> objects	Read-only. A list of the component color channels.
<code>countItems</code>	<a href="#">CountItems</a>	Read-only. The current count items.  <b>Note:</b> For additional information about count items, see Adobe Photoshop CS3 help on the Count Tool.
<code>fullName</code>	File	Read-only. The full path name of the document.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<code>height</code>	UnitValue	Read-only. The height of the document (unit value).  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<code>histogram</code>	array of 256 numbers (long)	Read-only. A histogram showing the number of pixels at each color intensity level for the composite channel.  <b>Note:</b> Valid only when <pre>mode = DocumentMode.RGB; mode = DocumentMode.CMYK; or mode = DocumentMode.INDEXEDCOLOR.</pre> See <a href="#">mode</a> .
<code>historyStates</code>	<a href="#">HistoryStates</a>	Read-only. The <code>HistoryStates</code> collection.
<code>info</code>	<a href="#">DocumentInfo</a>	Read-only. Metadata about the document.
<code>layerComps</code>	<a href="#">LayerComps</a>	Read-only. The <code>LayerComps</code> collection.
<code>layers</code>	<a href="#">Layers</a>	Read-only. The <code>Layers</code> collection.
<code>layerSets</code>	<a href="#">LayerSets</a>	Read-only. The <code>LayerSets</code> collection.
<code>managed</code>	boolean	Read-only. Indicates whether the document a is workgroup document.
<code>measurementScale</code>	<a href="#">MeasurementScale</a>	Read-only. The measurement scale for the document.  <b>Note:</b> The measurement scale feature is available in the Extended version only.
<code>mode</code>	<a href="#">DocumentMode</a>	Read-only. The color profile.
<code>name</code>	string	Read-only. The document's name.

Property	Value Type	What it is (Continued)
<b>parent</b>	<a href="#">Application</a>	Read-only. The <code>Document</code> object's container.
<b>path</b>	File	Read-only. The path to the document.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<b>pathItems</b>	<a href="#">PathItems</a>	Read-only. The <code>PathItems</code> collection.
<b>pixelAspectRatio</b>	number (double)	Read-write. The (custom) pixel aspect ratio to use (0.100 - 10.000).
<b>quickMaskMode</b>	boolean	Read-write. Indicates whether the document is in Quick Mask mode.
<b>resolution</b>	number (double)	Read-only. The document's resolution (in pixels per inch).
<b>saved</b>	boolean	Read-only. Indicates whether the document has been saved since the last change.
<b>selection</b>	<a href="#">Selection</a>	Read-only. The selected area of the document.
<b>typename</b>	string	Read-only. The class name of the <code>Document</code> object.
<b>width</b>	UnitValue	Read-only. The width of the document (unit value).  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<b>xmpMetadata</b>	<a href="#">xmpMetadata</a>	Read-only. Camera RAW settings for the image.  <b>Note:</b> Valid only for documents opened in Camera RAW format.

## Methods

Method	Parameter Type	Returns	What it does
<b>autoCount</b> (channel, threshold)	<a href="#">Channel</a> number (long)		Counts the number of objects in a document.  Creates a <a href="#">CountItem</a> object for each object counted.  <b>Note:</b> The <code>autoCount</code> feature is available in the Extended Version only.  For additional information about how to set up objects to count, please see the Count Tool in the Adobe Photoshop CS3 Help
<b>changeMode</b> (destinationMode [, options])	<a href="#">ChangeMode</a> ( <a href="#">BitmapConversionOptions</a> or <a href="#">IndexedConversionOptions</a> )		Changes the color profile.
<b>close</b> ([saving])	<a href="#">SaveOptions</a>		Closes the document. If any changes have been made, the script presents an alert with three options: save, do not save, prompt to save. The optional parameter specifies a selection in the alert box (default: <code>SaveOptions.PROMPTTOSAVECHANGES</code> ).
<b>convertProfile</b> (destinationProfile, intent [, blackPointCompensation] [, dither])	string <a href="#">Intent</a> boolean boolean		Changes the color profile.  <b>Note:</b> The <code>destinationProfile</code> parameter must be either a string that names the color mode or <code>Working RGB</code> , <code>Working CMYK</code> , <code>Working Gray</code> , <code>Lab Color</code> (meaning one of the working color spaces or Lab color).

Method	Parameter Type	Returns	What it does (Continued)
<b>crop</b> (bounds [, angle] [, width] [, height])	array(UnitValue) number (double) UnitValue UnitValue		Crops the document. The first parameter is an array of four coordinates that mark the portion remaining after cropping, in the following order: left, top, right, bottom.  <b>Note:</b> For information about the UnitValue type, see the <i>JavaScript Tools Guide</i> .
<b>duplicate</b> ([name ] [, mergeLayersOnly] )	string boolean	<a href="#">Document</a>	Creates a duplicate of the document object.  The optional parameter name provides the name for the duplicated document.  The optional parameter mergeLayersOnly indicates whether to only duplicate merged layers.
<b>exportDocument</b> (exportIn [, exportAs] [, options])	File <a href="#">ExportType</a> <a href="#">ExportOptionsIllustrator</a> or <a href="#">ExportOptionsSaveForWeb</a>		Exports the document.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>flatten</b> ()			Flattens all layers.
<b>flipCanvas</b> (direction)	<a href="#">Direction</a>		Flips the image within the canvas in the specified direction.
<b>importAnnotations</b> (file)	File		Imports annotations into the document.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>mergeVisibleLayers</b> ()			Flattens all visible layers in the document.

Method	Parameter Type	Returns	What it does (Continued)
<b>paste</b> ([intoSelection])	boolean	<a href="#">ArtLayer</a>	Pastes the contents of the clipboard into the document. If the optional argument is set to <code>true</code> and a selection is active, the contents are pasted into the selection.
<b>print</b> ([postScriptEncoding] [, sourceSpace] [, printSpace] [, intent] [blackPointCompensation])	<a href="#">PrintEncoding</a> <a href="#">SourceSpaceType</a> string <a href="#">Intent</a> boolean		Prints the document.  <b>Note:</b> <code>printSpace</code> specifies the color space for the printer. Valid values are <code>nothing</code> (that is, the same as the source); or <code>Working RGB</code> , <code>Working CMYK</code> , <code>Working Gray</code> , <code>Lab Color</code> (meaning one of the working color spaces or Lab color); or a string specifying a specific colorspace (default: <i>nothing</i> ).
<b>rasterizeAllLayers</b> ( )			Rasterizes all layers.
<b>recordMeasurements</b> ([source] [, dataPoints])	<a href="#">MeasurementSource</a> array of strings		Record measurements of document.
<b>resizeCanvas</b> ([width] [, height] [, anchor])	UnitValue UnitValue <a href="#">AnchorPosition</a>		Changes the size of the canvas to display more or less of the image but does not change the image size. See <a href="#">resizeImage</a> .  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>
<b>resizeImage</b> ([width] [, height] [, resolution] [, resampleMethod])	UnitValue UnitValue number (double) <a href="#">ResampleMethod</a>		Changes the size of the image.  <b>Note:</b> For information about the <code>UnitValue</code> object, see the <i>JavaScript Tools Guide</i>

Method	Parameter Type	Returns	What it does (Continued)
<b>revealAll</b> ( )			Expands the document to show clipped sections.
<b>rotateCanvas</b> (angle)	number (double)		Rotates the canvas (including the image) in clockwise direction.
<b>save</b> ( )			Saves the document.
<b>saveAs</b> (saveIn [, options]  [, asCopy] [, extensionType])	File formatSaveOptions object* boolean <a href="#">Extension</a>  * Examples: <a href="#">BMPSaveOptions</a> <a href="#">DCS2 SaveOptions</a> <a href="#">JPEGSaveOptions</a> <a href="#">TiffSaveOptions</a> etc.		Saves the document with specified save options.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>
<b>splitChannels</b> ( )		array of <a href="#">Document</a> objects	Splits the document channels into separate images.
<b>suspendHistory</b> (historyString javaScriptString)	string string		Provides a single entry in history states for the entire script provided by javaScriptString. Allows a single undo for all actions taken in the script.  The historyString parameter provides the string to use for the history state.  The javaScriptString parameter provides a string of JavaScript code to execute while history is suspended.
<b>trap</b> (width)	number (long)		Applies trapping to a CMYK document.  <b>Note:</b> Valid only when docRef.mode = DocumentMode.CMYK. See mode .

Method	Parameter Type	Returns	What it does (Continued)
<b>trim</b> ([type] [, top] [, left] [, bottom] [, right])	<a href="#">TrimType</a> boolean boolean boolean boolean		Trims the transparent area around the image on the specified sides of the canvas.  <b>Note:</b> Default is <code>true</code> for all boolean values.

## Sample Script

The following script creates a document that contains two images (a flower and a duck) obtained from the Adobe Photoshop CS3 Samples folder and employs the following steps:

- Determines which image is larger.
- Resizes the smaller image to match the larger image.
- Creates a merged document twice as high as either image in order to hold both images.
- Selects part of the document to and pastes the flower into the selection.
- Inverts the selection and pastes the duck into the lower part of the document.
- Positions the flower over the duck.

### Document.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS3 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

// first close all the open documents
while (app.documents.length) {
    app.activeDocument.close()
}

// Open the sunflower and duck files from the samples folder
var flowerDoc = open(File(app.path + "/Samples/Sunflower.psd"))
var duckDoc = open(File(app.path + "/Samples/Ducky.tif"))

// Find out which document is larger
// Resize the smaller document to the larger document's size
// The resize requires the document be the active/front document
if ((flowerDoc.width.value * flowerDoc.height.value) > (duckDoc.width.value *
    duckDoc.height.value)) {
    app.activeDocument = duckDoc
    duckDoc.resize(flowerDoc.width, flowerDoc.height)
} else {
    app.activeDocument = flowerDoc
    flowerDoc.resizeImage(duckDoc.width, duckDoc.height)
}

// Create a new document twice as high as two files
```



```
var mergedDoc = app.documents.add(duckDoc.width, duckDoc.height * 2,
duckDoc.resolution, "FlowerOverDuck")

// Copy the flower to the top; make it the active document so we can manipulate it
app.activeDocument = flowerDoc
flowerDoc.activeLayer.copy()

//Paste the flower to the merged document, making the merged document active
app.activeDocument = mergedDoc

// Select a square area at the top of the new document
var selRegion = Array(Array(0, 0),
    Array(mergedDoc.width.value, 0),
    Array(mergedDoc.width.value, mergedDoc.height.value / 2),
    Array(0, mergedDoc.height.value / 2),
    Array(0, 0))
// Create the selection
mergedDoc.selection.select(selRegion)

//Paste in the flower
mergedDoc.paste()

// do the same thing for the duck
app.activeDocument = duckDoc
duckDoc.activeLayer.copy()

app.activeDocument = mergedDoc
mergedDoc.selection.select(selRegion)

// Inverting the selection so the bottom of the document is now selected
mergedDoc.selection.invert()

// Paste the duck
mergedDoc.paste()

// get rid of our originals without modifying them
duckDoc.close(SaveOptions.DONOTSAVECHANGES)
flowerDoc.close(SaveOptions.DONOTSAVECHANGES)

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs
```

## DocumentInfo

Metadata about a `document` object. These values can be set by selecting File > File Info in the Adobe Photoshop CS3 application.

**Note:** Since the `DocumentInfo` class is also a property of the [Document](#) object, you use the property name `info`, rather than the class name `DocumentInfo`, in a script, as in the following sample, which sets the author, caption, and copyrighted properties:

```
var docRef = open(fileList[i])
// set the file info
docRef.info.author = "Mr. Adobe programmer"
docRef.info.caption = "Adobe Photo shoot"
docRef.info.copyrighted = CopyrightedType.COPYRIGHTEDWORK
```

The following sample uses the `DocumentInfo` object incorrectly:

```
docRef.DocumentInfo.author = "Mr. Adobe programmer"
docRef.DocumentInfo.caption = "Adobe Photo shoot"
docRef.DocumentInfo.copyrighted = CopyrightedType.COPYRIGHTEDWORK
```

## Properties

Property	Value Type	What it is
<code>author</code>	string	Read-write.
<code>authorPosition</code>	string	Read-write.
<code>caption</code>	string	Read-write.
<code>captionWriter</code>	string	Read-write.
<code>category</code>	string	Read-write.
<code>city</code>	string	Read-write.
<code>copyrighted</code>	<a href="#">CopyrightedType</a>	Read-write. The copyrighted status.
<code>copyrightNotice</code>	string	Read-write.
<code>country</code>	string	Read-write.
<code>creationDate</code>	string	Read-write.
<code>credit</code>	string	Read-write.
<code>exif</code>	array of arrays: Array(Array (tag, tag data)), ...)	Read-only. Camera data that includes camera settings used when the image was taken. Sample array values are: tag = "camera"; tag value = "Canon".
<code>headline</code>	string	Read-write.
<code>instructions</code>	string	Read-write.
<code>jobName</code>	string	Read-write.
<code>keywords</code>	array of strings	Read-write. A list of keywords that can identify the document or its contents.

Property	Value Type	What it is (Continued)
<code>ownerUrl</code>	string	Read-write.
<code>parent</code>	object ( <a href="#">Document</a> )	Read-only. The <code>info</code> object's container.
<code>provinceState</code>	string	Read-write.
<code>source</code>	string	Read-write.
<code>supplementalCategories</code>	array of strings	Read-write.
<code>title</code>	string	Read-write.
<code>transmissionReference</code>	string	Read-write.
<code>typename</code>	string	Read-only. The class name of the referenced <code>info</code> object.
<code>urgency</code>	<a href="#">Urgency</a>	Read-write.

## Sample Script

The following script sets document info (metadata) for all of the files in a specified folder and then saves the modified files as low-quality JPEG images in a new folder without changing the originals.

- Ask the user to specify the folder that contains the original files and the output folder for the JPEG images, and then check that the folders exist.
- Open each file and use the `documentInfo` object properties to tag it with the following metadata:
  - author: Mr. Adobe programmer
  - caption: Adobe Photo shoot
  - captionWriter: Mr. Adobe programmer
  - city: San Jose
  - copyrightNotice: Copyright (c) Adobe programmer Photography
  - copyrighted status: Copyrighted Work
  - country: USA
  - state: CA
- Save the new documents in JPEG format with a low quality setting.

### DocumentInfo.jsx

```
// Save the current preferences
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS3 to use pixels and display no dialogs
app.displayDialogs = DialogModes.NO

// ask the user for the input folder
var inputFolder = Folder.selectDialog("Select a folder to tag")

// ask the user for the output folder
var outputFolder = Folder.selectDialog("Select a folder for the output files")
```

```
// see if we got something interesting from the dialog
if (inputFolder != null && outputFolder != null) {

    // get all the files found in this folder
    var fileList = inputFolder.GetFiles()

    // save the outputs in JPEG
    var jpegOptions = new JPEGSaveOptions()

    // set the jpeg quality really low so the files are small
    jpegOptions.quality = 1

    // open each one in turn
    for (var i = 0; i < fileList.length; i++) {

        // The fileList includes both folders and files so open only files
        if (fileList[i] instanceof File && fileList[i].hidden == false) {

            // get a reference to the new document
            var docRef = open(fileList[i])

            // tag all of the documents with photo shoot information
            docRef.info.author = "Mr. Adobe programmer"
            docRef.info.caption = "Adobe Photo shoot"
            docRef.info.captionWriter = "Mr. Adobe programmer"
            docRef.info.city = "San Jose"
            docRef.info.copyrightNotice = "Copyright (c) Adobe programmer
                Photography"
            docRef.info.copyrighted = CopyrightedType.COPYRIGHTEDWORK
            docRef.info.country = "USA"
            docRef.info.provinceState = "CA"

            // change the date to a Adobe Photoshop CS3 date format
            // "YYYYMMDD"
            var theDate = new Date()

            // the year is from 1900 ???
            var theYear = (theDate.getYear() + 1900).toString()

            // convert the month from 0..12 to 00..12
            var theMonth = theDate.getMonth().toString()

            if (theDate.getMonth() < 10) {
                theMonth = "0" + theMonth
            }

            // convert the day from 0..31 to 00..31
            var theDay = theDate.getDate().toString()

            if (theDate.getDate() < 10) {
                theDay = "0" + theDay
            }

            // stick them all together
            docRef.info.creationDate = theYear + theMonth + theDay

            // flatten because we are saving to JPEG
            docRef.flatten()

            // go to 8 bit because we are saving to JPEG
```

```
        docRef.bitsPerChannel = BitsPerChannelType.EIGHT

        // save and close
        docRef.saveAs(new File(outputFolder + "/Output" + i + ".jpg"),
            jpegOptions)

        // don't modify the original
        docRef.close(SaveOptions.DONOTSAVECHANGES)
    }
}

// Reset the application preferences
app.displayDialogs = startDisplayDialogs
```

## Documents

The collection of open document objects. See [Document](#) for information on the document object.

**Note:** Because the `Documents` class is a property of the [Application](#) object, you use the property name, `documents`, rather than the class name, `Documents`, in your code, as in the following example:

```
documents.add(800, 500, 72, "myDocument", NewDocumentMode.RGB)
```

The following example, which uses an upper case *D* in the object name, is incorrect:

```
Documents.add(800, 500, 72, "myDocument", NewDocumentMode.RGB)
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>documents</code> collection.
<b>parent</b>	object ( <a href="#">Application</a> )	Read-only. The <code>documents</code> objects' container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>documents</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">Document</a>	Gets an element from the <code>documents</code> collection.
<b>add</b> ([width] [, height] [, resolution] [, name] [, mode] [, initialFill] [, pixelAspectRatio] [, bitsPerChannel]  [, colorProfileName])	UnitValue UnitValue number (double) string <a href="#">NewDocumentMode</a> <a href="#">DocumentFill</a> number (double) <a href="#">BitsPerChannelType</a> <a href="#">pe</a> string	<a href="#">Document</a>	Adds a document object.  pixelAspectRatio: range from 0.100 - 10.00. Default 1.0 for a square aspect ratio.  bitsPerChannelType has a default value of <code>BitsPerChannelType.EIGHT</code> .  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<b>getByName</b> (name)	string	<a href="#">Document</a>	Gets the first element in the <code>documents</code> collection with the provided name

# EPSOpenOptions

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

## Properties

Property	Value Type	What it is
<b>antiAlias</b>	boolean	Read-write. Indicates whether to use antialias.
<b>constrainProportions</b>	boolean	Read-write. Indicates whether to constrain the proportions of the image.
<b>height</b>	UnitValue	Read-write. The height of the image (unit value).  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i>
<b>mode</b>	<a href="#">OpenDocumentMode</a>	Read-write. The color profile to use as the document mode.
<b>resolution</b>	number (double)	Read-write. The resolution of the document in pixels per inch.
<b>typename</b>	string	Read-only. The class name of the referenced <code>EPSOpenOptions</code> object.
<b>width</b>	UnitValue	Read-write. The width of the image (unit value).  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i>

# EPSSaveOptions

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

## Properties

Property	Value Type	What it is
<code>embedColorProfile</code>	boolean	Read-write. Indicates whether to embed the color profile in this document.
<code>encoding</code>	<a href="#">SaveEncoding</a>	Read-write. The type of encoding to use (default: <code>SaveEncoding.BINARY</code> ).
<code>halftoneScreen</code>	boolean	Read-write. Indicates whether to include the halftone screen (default: <code>false</code> ).
<code>interpolation</code>	boolean	Read-write. Indicates whether to use image interpolation (default: <code>false</code> ).
<code>preview</code>	<a href="#">Preview</a>	Read-write. The preview type.
<code>psColorManagement</code>	boolean	Read-write. Indicates whether to use Postscript color management (default: <code>false</code> ).
<code>transferFunction</code>	boolean	Read-write. Indicates whether to include the Transfer functions to compensate for dot gain between the image and film (default: <code>false</code> ).
<code>transparentWhites</code>	boolean	Read-write. Indicates whether to display white areas as transparent.  <b>Note:</b> Valid only when <code>document.mode = DocumentMode.BITMAP</code> . See ' <a href="#">mode</a> ' on <a href="#">page 90</a> (in the Properties table of the <code>document</code> object) or ' <a href="#">changeMode</a> ' on <a href="#">page 92</a> (in the Methods table of the <code>document</code> object).
<code>typename</code>	string	Read-only. The class name of the referenced <code>EPSSaveOptions</code> object.
<code>vectorData</code>	boolean	Read-write. Indicates whether to include vector data.  <b>Note:</b> Valid only if the document includes vector data (text).



## ExportOptionsIllustrator

Options that can be specified when exporting a [PathItem](#) object to an Adobe Illustrator® file.

### Properties

Property	Value Type	What it is
<b>path</b>	<a href="#">IllustratorPathType</a>	Read-write. The type of path to export (default: <code>IllustratorPathType.DOCUMENTBOUNDS</code> ).
<b>pathName</b>	string	Read-write. The name of the path to export.  <b>Note:</b> Valid only when <code>path = IllustratorPathType.NAMEDPATH</code> . See <a href="#">path</a> .
<b>typename</b>	string	Read-only. The class name of the referenced <code>exportOptionsIllustrator</code> object.

# ExportOptionsSaveForWeb

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

## Properties

Property	Value type	What it is
<b>blur</b>	number (double)	Read-write. Applies blur to the image to reduce artifacts (default: 0.0).
<b>colorReduction</b>	<a href="#">ColorReductionType</a>	Read-write. The color reduction algorithm (default: <code>ColorReductionType.SELECTIVE</code> ).
<b>colors</b>	number (long)	Read-write. The number of colors in the palette (default: 256).
<b>dither</b>	<a href="#">Dither</a>	Read-write. The type of dither (default: <code>Dither.DIFFUSION</code> ).
<b>ditherAmount</b>	number (long)	Read-write. The amount of dither (default: 100).  <b>Note:</b> Valid only when <code>dither = Dither.DIFFUSION</code> . See <a href="#">dither</a> .
<b>format</b>	<a href="#">SaveDocumentType</a>	Read-write. The file format to use (default: <code>SaveDocumentType.COMPUSEVEGIF</code> ).  <b>Note:</b> For this property, only <code>COMPUSEVEGIF</code> , <code>JPEG</code> , <code>PNG-8</code> , <code>PNG-24</code> , and <code>BMP</code> are supported.
<b>includeProfile</b>	boolean	Read-write. Indicates whether to include the document's embedded color profile (default: <code>false</code> ).
<b>interlaced</b>	boolean	Read-write. Indicates whether to download in multiple passes; progressive (default: <code>false</code> ).
<b>lossy</b>	number (long)	Read-write. The amount of lossiness allowed (default: 0).
<b>matteColor</b>	<a href="#">RGBColor</a>	Read-write. The colors to blend transparent pixels against.
<b>optimized</b>	boolean	Read-write. Indicates whether to create smaller but less compatible files (default: <code>true</code> ).  <b>Note:</b> Valid only when <code>format = SaveDocumentType.JPEG</code> . See <a href="#">format</a> .

Property	Value type	What it is (Continued)
<b>PNG8</b>	boolean	Read-write. Indicates the number of bits; <code>true</code> = 8, <code>false</code> = 24 (default: <code>true</code> ).  <b>Note:</b> Valid only when <code>format = SaveDocumentType.PNG</code> . See <a href="#">format</a> .
<b>quality</b>	number (long)	Read-write. The quality of the produced image (0 - 100 as percentage; default: 60).
<b>transparency</b>	boolean	Read-write. Indication of transparent areas of the image should be included in the saved image (default: <code>true</code> ).
<b>transparencyAmount</b>	number (long)	Read-write. The amount of transparency dither (default: 100).  <b>Note:</b> Valid only if <code>transparency = true</code> . See <a href="#">transparency</a> .
<b>transparencyDither</b>	<a href="#">Dither</a>	Read-write. The transparency dither algorithm (default: <code>transparencyDither = Dither.NONE</code> ).
<b>typename</b>	string	Read-only. The class name of the referenced <code>ExportOptionsSaveForWeb</code> object.
<b>webSnap</b>	number (long)	Read-write. The tolerance amount within which to snap close colors to web palette colors (default: 0).

## GalleryBannerOptions

Options that define the `bannerOptions` property of the `galleryOptions` object. See [‘GalleryOptions’ on page 112](#).

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

### Properties

Property	Value Type	What it is
<code>contactInfo</code>	string	Read-write. The web photo gallery contact info.
<code>date</code>	string	Read-write. The web photo gallery date (default: current date).
<code>font</code>	<a href="#">GalleryFontType</a>	Read-write. The font setting for the banner text (default: <code>GalleryFontType.ARIAL</code> ).
<code>fontSize</code>	number (long)	Read-write. The font size for the banner text (1 - 7; default: 3).
<code>photographer</code>	string	Read-write. The web photo gallery photographer.
<code>siteName</code>	string	Read-write. The web photo gallery site name (default: <code>Adobe Web Photo Gallery</code> ).
<code>typename</code>	string	Read-only. The class name of the referenced <code>galleryBannerOptions</code> object.

## GalleryCustomColorOptions

Options that define the `customColorOptions` property of the `galleryOptions` object. See ['GalleryOptions' on page 112](#).

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

### Properties

Property	Value Type	What it is
<code>activeLinkColor</code>	<a href="#">RGBColor</a>	Read-write. The color to use to indicate an active link.
<code>backgroundColor</code>	<a href="#">RGBColor</a>	Read-write. The background color.
<code>bannerColor</code>	<a href="#">RGBColor</a>	Read-write. The banner color.
<code>linkColor</code>	<a href="#">RGBColor</a>	Read-write. The color to use to indicate a link.
<code>textColor</code>	<a href="#">RGBColor</a>	Read-write. The text color.
<code>typename</code>	string	Read-only. The class name of the referenced <code>galleryCustomColorOptions</code> object.
<code>visitedLinkColor</code>	<a href="#">RGBColor</a>	Read-write. The color to use to indicate a visited link.

# GalleryImagesOptions

Options that define the `imagesOptions` property of the `galleryOptions` object. See [‘GalleryOptions’ on page 112](#).

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

## Properties

Property	Value Type	What it is
<code>border</code>	number (long)	Read-write. The size (in pixels) of the border that separates images (0 - 99; default: 0).
<code>caption</code>	boolean	Read-write. Indication of whether to generate image captions (default: <code>false</code> ).
<code>dimension</code>	number (long)	Read-write. The resized image dimensions in pixels (default: 350).  <b>Note:</b> Valid only when <code>resizeImages = true</code> . See <a href="#">resizeImages</a> .
<code>font</code>	<a href="#">GalleryFontType</a>	Read-write. The font to use for image captions (default: <code>GalleryFontType.ARIAL</code> ).
<code>fontSize</code>	number (long)	Read-write. The font size for image captions (1 - 7; default: 3).  <b>Note:</b> Valid only when <code>caption = true</code> . See <a href="#">caption</a> .
<code>imageQuality</code>	number (long)	Read-write. The quality setting for a JPEG image (0 - 12; default: 5).
<code>includeCopyright</code>	boolean	Read-write. Indication of whether to include copyright information in captions (default: <code>false</code> ).  <b>Note:</b> Valid only when <code>caption = true</code> . See <a href="#">caption</a> .
<code>includeCredits</code>	boolean	Read-write. Indication of whether to include the credits in image captions (default: <code>false</code> ).  <b>Note:</b> Valid only when <code>caption = true</code> . See <a href="#">caption</a> .

Property	Value Type	What it is (Continued)
<code>includeFilename</code>	boolean	Read-write. Indication of whether to include the file name in image captions (default: <code>true</code> ).  <b>Note:</b> Valid only when <code>caption = true</code> . See <a href="#">caption</a> .
<code>includeTitle</code>	boolean	Read-write. Indication of whether to include the title in image captions (default: <code>false</code> ).  <b>Note:</b> Valid only when <code>caption = true</code> . See <a href="#">caption</a> .
<code>numericLinks</code>	boolean	Read-write. Indication of whether to add numeric links (default: <code>true</code> ).
<code>resizeConstraint</code>	<a href="#">GalleryConstrainType</a>	Read-write. The image dimensions to constrain in the gallery image (default: <code>GalleryConstrainType.CONSTRAINBOTH</code> ).  <b>Note:</b> Valid only when <code>resizeImages = true</code> . See <a href="#">resizeImages</a> .
<code>resizeImages</code>	boolean	Read-write. Indication of whether to automatically resize images for placement on the gallery pages (default: <code>true</code> ).
<code>typename</code>	string	Read-only. The class name of the referenced <code>galleryImagesOptions</code> object.

# GalleryOptions

Options that can be specified for a Web photo gallery.

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

## Properties

Property	Value Type	What it is
<code>addSizeAttributes</code>	boolean	Read-write. Indicates whether width and height attributes for images will be added (default: <code>true</code> ).
<code>bannerOptions</code>	<a href="#">GalleryBannerOptions</a>	Read-write. The options related to banner settings.
<code>customColorOptions</code>	<a href="#">GalleryCustomColorOptions</a>	Read-write. The options related to custom color settings.
<code>emailAddress</code>	string	Read-write. The email address to show on the web page.
<code>imagesOptions</code>	<a href="#">GalleryImagesOptions</a>	Read-write. The options related to images settings.
<code>includeSubFolders</code>	boolean	Read-write. Indication of whether to include all files found in sub folders of the input folder (default: <code>true</code> ).
<code>layoutStyle</code>	string	Read-write. The style to use for laying out the web page (default: <code>Centered Frame 1 - Basic</code> ).
<code>preserveAllMetadata</code>	boolean	Read-write. Indicates whether to save metadata (default: <code>false</code> ).
<code>securityOptions</code>	<a href="#">GallerySecurityOptions</a>	Read-write. The options related to security settings.
<code>thumbnailOptions</code>	<a href="#">GalleryThumbnailOptions</a>	Read-write. The options related to thumbnail image settings.
<code>typename</code>	string	Read-only. The class name of the referenced <code>galleryOptions</code> object.
<code>useShortExtension</code>	boolean	Read-write. Indicates whether the short web page extension <code>.htm</code> or number (long) web page extension <code>.html</code> will be used (default: <code>true</code> ).
<code>useUTF8Encoding</code>	boolean	Read-write. Indicates whether the web page should use UTF-8 encoding (default: <code>false</code> ).



## GallerySecurityOptions

Options that define the `securityOptions` property of the `galleryOptions` object. See ['GalleryOptions' on page 112](#).

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

### Properties

Property	Value Type	What it is
<code>content</code>	<a href="#">GallerySecurityType</a>	Read-write. The web photo gallery security content (default: <code>GallerySecurityType.NONE</code> ).
<code>font</code>	<a href="#">GalleryFontType</a>	Read-write. The web photo gallery security font (default: <code>GalleryFontType.ARIAL</code> ).
<code>fontSize</code>	number (long)	Read-write. The web photo gallery security font size (1 - 72; default: 3).
<code>opacity</code>	number (long)	Read-write. The web page security opacity as a percent (default: 100).
<code>text</code>	string	Read-write. The web photo gallery security custom text.
<code>textColor</code>	<a href="#">RGBColor</a>	Read-write. The web page security text color.
<code>textPosition</code>	<a href="#">GallerySecurityTextPositionType</a>	Read-write. The web photo gallery security text position (default: <code>GallerySecurityTextPositionType.CENTERED</code> ).
<code>textRotate</code>	<a href="#">GallerySecurityTextRotateType</a>	Read-write. The web photo gallery security text orientation to use (default: <code>GallerySecurityTextRotateType.ZERO</code> ).
<code>typename</code>	string	Read-only. The class name of the referenced <code>gallerySecurityOptions</code> object.

# GalleryThumbnailOptions

Options that define the `thumbnailOptions` property of the `galleryOptions` object. See [‘GalleryOptions’ on page 112](#).

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

## Properties

Property	Value Type	What it is
<b>border</b>	number (long)	Read-write. The amount of border pixels you want around your thumbnail images (0 - 99; default: 0).
<b>caption</b>	boolean	Read-write. Indicates whether there is a caption (default: <code>false</code> ).
<b>columnCount</b>	number (long)	Read-write. The number of columns on the page (default: 5).
<b>dimension</b>	number (long)	Read-write. The web photo gallery thumbnail dimension in pixels (default: 75).
<b>font</b>	<a href="#">GalleryFontType</a>	Read-write. The web photo gallery font (default: <code>GalleryFontType.ARIAL</code> ).
<b>fontSize</b>	number (long)	Read-write. The font size for thumbnail images text (1 - 7; default: 3).
<b>includeCopyright</b>	boolean	Read-write. Indication of whether to include copyright information for thumbnails (default: <code>false</code> ).
<b>includeCredits</b>	boolean	Read-write. Indication of whether to include credits for thumbnails (default: <code>false</code> ).
<b>includeFilename</b>	boolean	Read-write. Indication of whether to include file names for thumbnails (default: <code>false</code> ).
<b>includeTitle</b>	boolean	Read-write. Indication of whether to include titles for thumbnails (default: <code>false</code> ).
<b>rowCount</b>	number (long)	Read-write. The number of rows on the page (default: 3).
<b>size</b>	<a href="#">GalleryThumbSizeType</a>	Read-write. The thumbnail image size (default: <code>GalleryThumbSizeType.MEDIUM</code> ).
<b>typename</b>	string	Read-only. The class name of the referenced <code>GalleryThumbnailOptions</code> object.

# GIFSaveOptions

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

## Properties

Property	Value Type	What it is
<b>colors</b>	number (long)	Read-write. The number of palette colors.  <b>Note:</b> Valid only when palette = Palette.LOCALADAPTIVE; palette = Palette.LOCALPERCEPTUAL; palette = Palette.LOCALSELECTIVE; palette = Palette.MACOSPALETTE; palette = Palette.UNIFORM; palette = Palette.WEBPALETTE; or palette = Palette.WINDOWSPALETTE. See <a href="#">palette</a> .
<b>dither</b>	<a href="#">Dither</a>	Read-write. The dither type.
<b>ditherAmount</b>	number (long)	Read-write. The amount of dither. (1 - 100; default: 75).  <b>Note:</b> Valid only when dither = Dither.DIFFUSION. See <a href="#">dither</a> .
<b>forced</b>	<a href="#">ForcedColors</a>	Read-write. The type of colors to force into the color palette.
<b>interlaced</b>	boolean	Read-write. Indicates whether rows should be interlaced (default: false).
<b>matte</b>	<a href="#">MatteType</a>	Read-write. The color to use to fill anti-aliased edges adjacent to transparent areas of the image (default: MatteType.WHITE).  <b>Note:</b> When transparency = false, the matte color is applied to transparent areas. See <a href="#">transparency</a> .
<b>palette</b>	<a href="#">Palette</a>	Read-write. The type of palette to use (default: Palette.LOCALSELECTIVE).
<b>preserveExactColors</b>	boolean	Read-write. Indicates whether to protect colors in the image that contain entries in the color table from being dithered.  <b>Note:</b> Valid only when dither = Dither.DIFFUSION. See <a href="#">dither</a> .

Property	Value Type	What it is (Continued)
<b>transparency</b>	boolean	Read-write. Indicates whether to preserve transparent areas of the image during conversion to GIF format.
<b>typename</b>	string	Read-only. The class name of the referenced <code>GIFSaveOptions</code> object.

# GrayColor

Options for defining a gray color.

## Properties

Property	Value Type	What it is
<b>gray</b>	number (double)	Read-write. The gray value (0.0 - 100.0; default: 0.0).
<b>typename</b>	string	Read-only. The class name of the referenced <code>grayColor</code> object.

## HistoryState

A version of the document stored automatically (and added to the `HistoryStates` collection), which preserves the document's state, each time the document is changed. See [HistoryStates](#) for information about the `HistoryStates` collection.

**Note:** Because the `HistoryState` class is also a property of the [Document](#) object, you use the property name, `historyState`, rather than the class name, `HistoryState`, in your code.

The following example uses correct syntax to refer to a `HistoryState` object named `AddLayerMask` and then assign its `snapshot` property value:

```
documents(0).historyState("AddLayerMask").snapshot = true
```

The following example, which uses an upper case A in the object name, is incorrect:

```
documents(0).HistoryState("AddLayerMask").snapshot = true
```

## Properties

Property	Value Type	What it is
<b>name</b>	string	Read-only. The <code>HistoryState</code> object's name.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The <code>HistoryState</code> object's container.
<b>snapshot</b>	boolean	Read-only. Indicates whether the history state is a snapshot.
<b>typename</b>	string	Read-only. The class name of the referenced <code>HistoryState</code> object.

# HistoryStates

The collection of `HistoryState` objects in the document. See [HistoryState](#) for more information on `HistoryState` objects.

**Note:** Because the `HistoryStates` class is also a property of the [Document](#) object, you use the property name, `historyStates`, rather than the class name, `HistoryStates`, in your code.

The following example uses correct syntax to fill a `Selection` object (referred to by the variable `selRef`) with an object in the `HistoryStates` collection:

```
selRef.fill(activeDocument.historyStates[7])
```

The following example, which uses an upper case *H* in the object name, is incorrect:

```
selRef.fill(activeDocument.HistoryStates[7])
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>HistoryStates</code> collection.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The <code>HistoryStates</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>HistoryStates</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">HistoryState</a>	Gets an element from the <code>HistoryStates</code> collection.
<b>getByName</b> (name)	string	<a href="#">HistoryState</a>	Get the first element in the <code>HistoryStates</code> collection with the provided name.

# HSBColor

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

## Properties

Property	Value Type	What it is
<b>brightness</b>	number (double)	Read-write. The brightness value (between 0.0 and 100.0).
<b>hue</b>	number (double)	Read-write. The hue value (between 0.0 and 360.0).
<b>saturation</b>	number (double)	Read-write. The saturation value (between 0.0 and 100.0).
<b>typename</b>	string	Read-only. The class name of the referenced <code>HSBColor</code> object.



# IndexedConversionOptions

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

## Properties

Property	Value Type	What it is
<b>colors</b>	number (long)	Read-write. The number of palette colors.  <b>Note:</b> Valid only when <code>palette = Palette.LOCALADAPTIVE;</code> <code>palette = Palette.LOCALPERCEPTUAL;</code> <code>palette = Palette.LOCALSELECTIVE;</code> <code>palette = Palette.MACOSPALETTE;</code> <code>palette = Palette.UNIFORM;</code> <code>palette = Palette.WEBPALETTE;</code> or <code>palette = Palette.WINDOWSPALETTE .</code> See <a href="#">palette</a> .
<b>dither</b>	<a href="#">Dither</a>	Read-write. The dither type.
<b>ditherAmount</b>	number (long)	Read-write. The amount of dither. (1 - 100).  <b>Note:</b> Valid only when <code>dither = Dither.diffusion</code> .
<b>forced</b>	<a href="#">ForcedColors</a>	Read-write. The type of colors to force into the color palette.
<b>matte</b>	<a href="#">MatteType</a>	Read-write. The color to use to fill anti-aliased edges adjacent to transparent areas of the image (default: <code>MatteType.WHITE</code> ).  <b>Note:</b> When <code>transparency = false</code> , the matte color is applied to transparent areas. See <a href="#">transparency</a> .
<b>palette</b>	<a href="#">Palette</a>	Read-write. The palette type (default: <code>Palette.EXACT</code> ).
<b>preserveExactColors</b>	boolean	Read-write. Indicates whether to protect colors in the image that contain entries in the color table from being dithered.  <b>Note:</b> Valid only when <code>dither = Dither.DIFFUSION</code> . See <a href="#">dither</a> .
<b>transparency</b>	boolean	Read-write. Indicates whether to preserve transparent areas of the image during conversion to GIF format.
<b>typename</b>	string	Read-only. The class name of the referenced <code>IndexedConversionOptions</code> object.

# JPEGSaveOptions

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

## Properties

Property	Value Type	What it is
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<b>formatOptions</b>	<a href="#">FormatOptions</a>	Read-write. The download format to use (default: <code>FormatOptions.STANDARDBASELINE</code> ).
<b>matte</b>	<a href="#">MatteType</a>	Read-write. The color to use to fill anti-aliased edges adjacent to transparent areas of the image (default: <code>MatteType.WHITE</code> ).  <b>Note:</b> When <code>transparency = false</code> , the matte color is applied to transparent areas. See <a href="#">transparency</a> .
<b>quality</b>	number (long)	Read-write. The image quality setting to use (affects file size and compression) (0 - 12; default: 3).
<b>scans</b>	number (long)	Read-write. The number of scans to make to incrementally display the image on the page (3 - 5; default: 3).  <b>Note:</b> Valid only for when <code>formatOptions = FormatOptions.PROGRESSIVE</code> .
<b>typename</b>	string	Read-only. The class name of the referenced <code>JPEGSaveOptions</code> object.

# LabColor

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

## Properties

Property	Value Type	What it is
<b>a</b>	number (double)	Read-write. The a-value (-128.0 - 127.0).
<b>b</b>	number (double)	Read-write. The b-value (-128.0 - 127.0).
<b>l</b>	number (double)	Read-write. The L-value (0.0 - 100.0).
<b>typename</b>	string	Read-only. The class name of the referenced <code>LabColor</code> object.

# LayerComp

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

**Note:** Because the `LayerComp` class is also a property of the [Document](#) object, you use the property name, `layerComp`, rather than the class name, `LayerComp`, in your code.

The following example uses correct syntax to set the comment property value for a `LayerComp` object named `myLayerComp`:

```
activeDocument.layerComp("myLayerComp").comment = "View from shoreline"
```

The following example, which uses an upper case *L* in the object name, is incorrect:

```
activeDocument.LayerComp("myLayerComp").comment = "View from shoreline"
```

## Properties

Property	Value Type	What it is
<b>appearance</b>	boolean	Read-write. Indicates whether to use layer appearance (layer styles) settings.
<b>comment</b>	string	Read-write. A description of the layer comp.
<b>name</b>	string	Read-write. The name of the layer comp.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-write. The <code>layerComp</code> object's container.
<b>position</b>	boolean	Read-write. Indicates whether to use layer position.
<b>selected</b>	boolean	Read-only. Indicates whether the layer comp is currently selected.
<b>typename</b>	string	Read-only. The class name of the referenced <code>layerComp</code> object.
<b>visibility</b>	boolean	Read-write. Indicates whether to use layer visibility settings .

## Methods

Method	Parameter Type	Returns	What it does
<b>apply</b> ( )			Applies the layer comp to the document.
<b>recapture</b> ( )			Recaptures the current layer state(s) for this layer comp.

Method	Parameter Type	Returns	What it does (Continued)
<b>remove</b> ( )			Deletes the <code>layerComp</code> object.
<b>resetfromComp</b> ( )			Resets the layer comp state to the document state.

## LayerComps

The collection of `layerComp` objects in the document. See [LayerComp](#) for information on `layerComp` objects.

**Note:** Because the `LayerComps` class is also a property of the [Document](#) object, you use the property name, `layerComps`, rather than the class name, `LayerComps`, in your code.

The following example uses correct syntax to add a `LayerComps`:

```
activeDocument.layerComps.add("myLayerComp", "View from Shoreline", true, true, true)
```

The following example, which uses an upper case `L` in the object name, is incorrect:

```
activeDocument.LayerComps.add("myLayerComp", "View from Shoreline", true, true, true)
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>layerComps</code> collection.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The <code>layerComps</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>layerComps</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">LayerComp</a>	Gets an element from the <code>layerComps</code> collection.
<b>add</b> (name, comment, appearance, position, visibility)	string string boolean boolean boolean	<a href="#">LayerComp</a>	Adds a layer comp.
<b>getByName</b> (name)	string	<a href="#">LayerComp</a>	Gets the first element in the collection with the provided name.
<b>removeAll</b> ( )			Removes all <code>layerComp</code> objects from the <code>layerComps</code> collection.

# Layers

The collection of layer objects, including [ArtLayer](#) and [LayerSet](#) objects, in the document.

**Note:** Because the `Layers` object is a property of the [Document](#) object (as well as several other objects), you use the property name, `layers`, rather than the class name, `Layers`, in your code. The following example uses the `length` property to count the number of `layer` objects in the active document, then displays the number on the screen:

```
var layerNum = app.activeDocument.layers.length
alert(layerNum)
```

The following example uses an upper case `L`, which is incorrect:

```
var layerNum = app.activeDocument.Layers.length
alert(layerNum)
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>layers</code> collection.
<b>parent</b>	object (document or layerSet)	Read-only. The <code>layers</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>layers</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	object (Layer)	Gets an element from the collection.
<b>getByName</b> (name)	string	Layer	Gets the first element in the <code>layers</code> collection with the provided name.
<b>removeAll</b> ( )			Removes all layers from the collection.

# LayerSet

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

**Note:** Most likely, you will use variables to refer to `layerSet` objects in your script. However, if you choose not to use a variable, be aware that, because the `LayerSet` class is also a property of the [Document](#) object, you use the property name, `layerSet`, rather than the class name, `LayerSet`, in your code.

The following example uses correct syntax to refer to a `layerSet` object by name and then assign its `allLocked` property value:

```
documents(0).layerSet("myLayerSet").allLocked = true
```

The following example, which uses an upper case *L* in the object name, is incorrect:

```
documents(0).LayerSet("myLayerSet").allLocked = true
```

## Properties

Property	Value Type	What it is
<b>allLocked</b>	boolean	Read-write. Indicates whether the contents in the layers contained in the <code>layerSet</code> object are editable.
<b>artLayers</b>	<a href="#">ArtLayers</a>	Read-only. The <code>artLayer</code> objects in this layer set.
<b>blendMode</b>	<a href="#">BlendMode</a>	Read-write. The blend mode to use for the layer set.
<b>bounds</b>	array (UnitValue)	Read-only. The bounding rectangle of the layer set.  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<b>enabledChannels</b>	array of <a href="#">Channel</a> objects	Read-write. The channels enabled for the layer set; must be a list of component channels.  <b>Note:</b> See <code>kind</code> in the Properties table for the <code>Channel</code> object ( <a href="#">Channel</a> ).
<b>layers</b>	<a href="#">Layers</a>	Read-only. The layers in this <code>layerSet</code> object.
<b>layerSets</b>	<a href="#">LayerSets</a>	Read-only. Layer Sets contained within a Layer Set.
<b>linkedLayers</b>	array of <a href="#">ArtLayer</a> and/or <a href="#">LayerSet</a>	Read-only. The layers linked to this <code>layerSet</code> object.
<b>name</b>	string	Read-write. The name of the <code>layerSet</code> object.
<b>opacity</b>	number (double)	Read-write. The master opacity of the <code>layerSet</code> object (0.0 - 100.0).
<b>parent</b>	object ( <a href="#">Document</a> or <a href="#">LayerSet</a> )	Read-only. The <code>layerSet</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>layerSet</code> object.
<b>visible</b>	boolean	Read-write. Indicates whether the <code>layerSet</code> object is visible.



## Methods

Method	Parameter Type	Returns	What it does
<b>duplicate</b> ([relativeObject]  [, insertionLocation])	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> ) <a href="#">ElementPlacement</a>	<a href="#">LayerSet</a>	Creates a duplicate of the layerSet object.
<b>link</b> (with)	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> )		Links the layer set with another layer.
<b>merge</b> ( )		<a href="#">ArtLayer</a>	Merges the layerset; returns a reference to the art layer created by this method.
<b>move</b> (relativeObject,  insertionLocation)	object ( <a href="#">ArtLayer</a> or <a href="#">LayerSet</a> ) <a href="#">ElementPlacement</a>		Moves the layerSet object.
<b>remove</b> ( )			Deletes the layerSet object.
<b>resize</b> ([horizontal] [, vertical] [, anchor])	number (double) number (double) <a href="#">AnchorPosition</a>		Resizes all layers in the layer set to the specified dimensions (as a percentage of its current size) and places the layer set in the specified position.
<b>rotate</b> (angle [, anchor])	number (double) <a href="#">AnchorPosition</a>		Rotates all layers in the layer set around the specified anchor point (default: <code>AnchorPosition.MIDDLECENTER</code> ).
<b>translate</b> ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the position relative to its current position.  <b>Note:</b> For more information about the UnitValue type, see the <i>JavaScript Tools Guide</i> .
<b>unlink</b> ( )			Unlinks the layer set.

# LayerSets

The collection of `layerSet` objects in the document. See [LayerSet](#) for information on `layerSet` objects.

**Note:** Because the `LayerSets` class is a property of the [Document](#) object, you use the property name, `layerSets`, rather than the class name, `LayerSets`, in your code. For example:

```
var layerSetRef = docRef.layerSets.add()
```

The following sample uses the `layerSets` object incorrectly:

```
var layerSetRef = docRef.LayerSets.add()
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>LayerSets</code> collection.
<b>parent</b>	object ( <a href="#">Document</a> or <a href="#">LayerSet</a> )	Read-only. The <code>layerSets</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>layerSets</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">LayerSet</a>	Gets an element from the <code>layerSets</code> collection.
<b>add</b> ( )		<a href="#">LayerSet</a>	Creates a new <code>layerSet</code> object.
<b>getByName</b> (name)	string	<a href="#">LayerSet</a>	Gets the first element in the <code>layerSets</code> collection with the provided name.
<b>removeAll</b> ( )			Removes the layer set, and any layers or layer sets it contains, from the document.

## Sample Script

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

**Note:** The script uses the ExtendScript `$` object. For further details, see the *JavaScript Tools Guide*.

### LayerSets.jsx

```
$.level = 1

//close all open documents
while (app.documents.length) {
    app.activeDocument.close()
}
```

```
// create a working document
var docRef = app.documents.add()

// create an array to hold the layer sets
var myLayerSets = new Array()

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

//Create an indexer variable
var i = 0

// Create three layer sets at the top level
for (i = 0; i < 3; i++) {
    myLayerSets[i] = new Array()
    myLayerSets[i][0] = docRef.layerSets.add()
}

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

// Create a layer set inside each layer set
for (i = 0; i < 3; i++) {
    myLayerSets[i][0].name = textArray[i] + " Set"
    myLayerSets[i][1] = myLayerSets[i][0].layerSets.add()
    myLayerSets[i][1].name = "Inside " + textArray[i] + " Set"
}

// Create an array to hold the layers
var myLayers = new Array()

// Create a text layer with a description inside each layer set
for (i = 0; i < 3; i++) {
    myLayers[i] = myLayerSets[i][1].artLayers.add()
    myLayers[i].kind = LayerKind.TEXT
    myLayers[i].textItem.contents = "Layer in " + textArray[i] + " Set Inside "
        + textArray[i] + " Set"
    myLayers[i].textItem.position = Array(app.activeDocument.width * i * 0.33,
        app.activeDocument.height * (i + 1) * 0.25)
    myLayers[i].textItem.size = 12
}
```

# MeasurementLog

The measurement log for the application. See [measurementLog](#) (in the Properties table for the [Application](#) object.)

**Note:** The MeasurementLog feature is available in the Extended Version only.

Because the `MeasurementLog` class is a property of the `Application` object, you use the property name, `measurementLog`, rather than the class name, `MeasurementLog`, in your code.

## Methods

Method	Parameter type	Returns	What it does
<b>exportMeasurements</b> ( [file] [, range] [, dataPoints] )	File <a href="#">MeasurementRange</a> array of strings		Export some measurement(s).
<b>deleteMeasurements</b> ( [range] )	<a href="#">MeasurementRange</a>		Delete a measurement.

# MeasurementScale

The measurement scale for the document. See [measurementScale](#) (in the Properties table for the [Document](#) object.)

**Note:** The MeasurementScale feature is available in the Extended Version only.

Because the `MeasurementScale` class is a property of the `Document` object, you use the property name, `measurementScale`, rather than the class name, `MeasurementScale`, in your code. For example:

```
activeDocument.measurementScale.pixelLength = 25
```

The following code incorrectly uses an upper case *M*:

```
activeDocument.MeasurementScale.pixelLength = 25
```

## Properties

Property	Value Type	What it is
<code>pixelLength</code>	number (long)	Read-write. The length in pixels this scale equates to.
<code>logicalLength</code>	number (double)	Read-write. The logical length this scale equates to.
<code>logicalUnits</code>	string	Read-write. The logical units for this scale.

# NoColor

An object that represents a missing color.

## Properties

Property	Value type	What it is
<code>typename</code>	string	Read-only. The class name of the referenced <code>noColor</code> object.

# Notifier

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

**Note:** For notifiers to work, they must be enabled. See the [notifiersEnabled](#) property of the Application object.

**Note:** Events that occur within scripts do not generally trigger notifiers, because they occur inside of a “play script” event.

**Note:** Because the `Notifier` class is also a property of the [Application](#) object, you use the property name, `notifier`, rather than the class name, `Notifier`, in your code.

## 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.  <b>Note:</b> For a list of four-character codes, see <a href="#">Appendix A: Event ID Codes</a> .
<code>eventClass</code>	string	Read-only. The class ID associated with the <code>event</code> for the <code>Notifier</code> object, four characters or a unique string.  <b>Note:</b> When an event applies to multiple types of objects, you use this property to distinguish which object this <code>Notifier</code> applies to. For example, the Make event (“Mk ”) applies to documents (“Dcmn”), channels (“Chnl”) and other objects.
<code>eventFile</code>	File	Read-only. The path to the file to execute when the event occurs/activates the notifier.  <b>Note:</b> For information about the <code>File</code> object, see the <i>JavaScript Tools Guide</i>
<code>parent</code>	object ( <a href="#">Application</a> )	Read-only. The <code>notifier</code> object’s container.
<code>typename</code>	string	Read-only. The class name of the referenced <code>notifier</code> object.

## Methods

Method	Parameter type	Returns	What it does
<code>remove</code> <code>()</code>			<p>Deletes the <code>notifier</code> object.</p> <p><b>Note:</b> You can remove a <code>notifier</code> object from the Script Events Manager drop-down list by deleting the file named <code>Script Events Manager.xml</code> from in the Photoshop preferences folder. See Adobe Photoshop CS3 help for more information.</p>



## Notifiers

The collection of `notifier` objects in the document; the `notifiers` property of the `app` object. See [‘Notifier’ on page 135](#) for information on `notifier` objects. See [notifiers](#) (in the Properties table of the `app` object).

In order to enable notifiers to run scripts

**Note:** Because the `Notifiers` class is a property of the [Application](#) object, you use the property name, `notifiers`, rather than the class name, `Notifiers`, in your code. For example:

```
var notRef = app.notifiers.add("OnClickGoButton", eventFile)
```

The following sample uses the `Notifiers` object incorrectly:

```
var notRef = app.Notifiers.add("OnClickGoButton", eventFile)
```

## Properties

Property	Value type	What it is
<code>length</code>	number (long)	Read-only. The number of elements in the <code>notifiers</code> collection.
<code>parent</code>	object ( <a href="#">Application</a> )	Read-only. The <code>notifiers</code> object's container
<code>typename</code>	string	Read-only. The class name of the referenced <code>notifiers</code> object.

## Methods

Method	Parameter type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">Notifier</a>	Gets an element from the <code>notifiers</code> collection.
<b>add</b> (event, eventFile [, eventClass])	string File string	<a href="#">Notifier</a>	<p>Creates a <code>Notifier</code> object.</p> <p><code>event</code> defines the class ID of the event: four characters or a unique string. For a list of four-character codes, see <a href="#">Appendix A: Event ID Codes</a>.</p> <p><b>Tip:</b> Remember to omit the single quotes when including a four-character ID in your code.</p> <p><code>eventFile</code> defines the script file that executes when the event occurs.</p> <p><b>Note:</b> An <code>eventClass</code> value corresponds to the class of object the event is applied to: four characters or a unique string. When an <code>event</code> applies to multiple types of objects, you use the <code>eventClass</code> parameter to distinguish which object this <code>Notifier</code> applies to. For example, the Make event ("Mk ") applies to documents ("Dcmn"), channels ("Chnl") and other objects.</p>
<b>removeAll</b> ( )			<p>Removes all <code>Notifier</code> objects from the <code>notifiers</code> collection.</p> <p><b>Note:</b> You can remove a <code>Notifier</code> object from the Script Events Manager drop-down list by deleting the file named <code>Script Events Manager.xml</code> from in the Photoshop preferences folder. See Adobe Photoshop CS3 help for more information.</p>

## PathItem

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

**Note:** Because the `PathItem` class is also a property of the [Document](#) object, you use the property name, `pathItem`, rather than the class name, `PathItem`, in your code.

The following example uses correct syntax to select a `pathItem` object :

```
activeDocument.pathItem("myPath").select()
```

The following example, which uses an upper case *P* in the object name, is incorrect:

```
activeDocument.PathItem("myPath").select()
```

## Properties

Property	Value Type	What it is
<b>kind</b>	<a href="#">PathKind</a>	Read-write. The <code>pathItem</code> object's type.
<b>name</b>	string	Read-write. The <code>pathItem</code> object's name.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The <code>pathItem</code> object's container.
<b>SubPathItems</b>	<a href="#">SubPathItems</a>	Read-only. The sub path objects for this <code>pathItem</code> object.
<b>typename</b>	string	Read-only. The class name of the referenced <code>pathItem</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>deselect</b> ( )			Deselects this <code>pathItem</code> object.
<b>duplicate</b> (name)	string		Duplicates this <code>pathItem</code> object with the new name specified in the argument.
<b>fillPath</b> ([fillColor]  [, mode] [, opacity] [, preserveTransparency] [, feather] [, wholePath] [, antiAlias])	Object ( <a href="#">SolidColor</a> , <a href="#">ArtLayer</a> , <a href="#">HistoryState</a> ) <a href="#">ColorBlendMode</a> number (double) boolean number (double) boolean boolean		Fills the area enclosed by the path (opacity: 0 - 100 as percent; feather: 0.0 - 250.0 in pixels).  The <code>wholePath</code> parameter indicates that all subpaths are used when doing the fill. (Default: <code>true</code> ).

Method	Parameter Type	Returns	What it does (Continued)
<b>makeClippingPath</b> ([flatness])	number (double)		Makes this <code>pathItem</code> object the clipping path for this document; the optional parameter tells the PostScript printer how to approximate curves in the path (0.2 - 100).
<b>makeSelection</b> ([feather] [, antiAlias] [, operation])	number (double) boolean <a href="#">SelectionType</a>		Makes a <code>selection</code> object, whose border is the path, from this <code>pathItem</code> object (feather: 0.0 - 250.0 in pixels).  <b>Note:</b> See <a href="#">Selection</a> .
<b>remove</b> ( )			Deletes this <code>pathItem</code> object.
<b>select</b> ( )			Makes this <code>pathItem</code> object the active or selected <code>pathItem</code> object.
<b>strokePath</b> ([tool] [, simulatePressure])	<a href="#">ToolType</a> boolean		Strokes the path with the specified information.

## Sample Script

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

### Paths.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS3 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

// first close all the open documents
while (app.documents.length) {
    app.activeDocument.close()
}

// create a document to work with
var docRef = app.documents.add(5000, 7000, 72, "Simple Line")

//line 1--it's a straight line so the coordinates for anchor, left, and right
//for each point have the same coordinates
var lineArray = new Array()
    lineArray[0] = new PathPointInfo
    lineArray[0].kind = PointKind.CORNERPOINT
```

```
lineArray[0].anchor = Array(100, 100)
lineArray[0].leftDirection = lineArray[0].anchor
lineArray[0].rightDirection = lineArray[0].anchor

lineArray[1] = new PathPointInfo
lineArray[1].kind = PointKind.CORNERPOINT
lineArray[1].anchor = Array(150, 200)
lineArray[1].leftDirection = lineArray[1].anchor
lineArray[1].rightDirection = lineArray[1].anchor

var lineSubPathArray = new Array()
lineSubPathArray[0] = new SubPathInfo()
lineSubPathArray[0].operation = ShapeOperation.SHAPEXOR
lineSubPathArray[0].closed = false
lineSubPathArray[0].entireSubPath = lineArray

// line 2
var lineArray2 = new Array()
lineArray2[0] = new PathPointInfo
lineArray2[0].kind = PointKind.CORNERPOINT
lineArray2[0].anchor = Array(150, 200)
lineArray2[0].leftDirection = lineArray2[0].anchor
lineArray2[0].rightDirection = lineArray2[0].anchor

lineArray2[1] = new PathPointInfo
lineArray2[1].kind = PointKind.CORNERPOINT
lineArray2[1].anchor = Array(200, 100)
lineArray2[1].leftDirection = lineArray2[1].anchor
lineArray2[1].rightDirection = lineArray2[1].anchor

lineSubPathArray[1] = new SubPathInfo()
lineSubPathArray[1].operation = ShapeOperation.SHAPEXOR
lineSubPathArray[1].closed = false
lineSubPathArray[1].entireSubPath = lineArray2

//ice cream curve
//it's a curved line, so there are 3 points, not 2
//coordinates for the middle point (lineArray3[1]) are different.
//The left direction is positioned "above" the anchor on the screen.
//The right direction is positioned "below" the anchor
//You can change the coordinates for these points to see
//how the curve works...
var lineArray3 = new Array()
lineArray3[0] = new PathPointInfo
lineArray3[0].kind = PointKind.CORNERPOINT
lineArray3[0].anchor = Array(200, 100)
lineArray3[0].leftDirection = lineArray3[0].anchor
lineArray3[0].rightDirection = lineArray3[0].anchor

lineArray3[1] = new PathPointInfo
lineArray3[1].kind = PointKind.CORNERPOINT
lineArray3[1].anchor = Array(150, 50)
lineArray3[1].leftDirection = Array(100, 50)
lineArray3[1].rightDirection = Array(200, 50)

lineArray3[2] = new PathPointInfo
lineArray3[2].kind = PointKind.CORNERPOINT
lineArray3[2].anchor = Array(100, 100)
lineArray3[2].leftDirection = lineArray3[2].anchor
lineArray3[2].rightDirection = lineArray3[2].anchor
```

```
lineSubPathArray[2] = new SubPathInfo()
lineSubPathArray[2].operation = ShapeOperation.SHAPEXOR
lineSubPathArray[2].closed = false
lineSubPathArray[2].entireSubPath = lineArray3

//create the path item
var myPathItem = docRef.pathItems.add("A Line", lineSubPathArray)

// stroke it so we can see something
myPathItem.strokePath(ToolType.BRUSH)

// Reset the application preferences
preferences.rulerUnits = startRulerUnits
preferences.typeUnits = startTypeUnits
displayDialogs = startDisplayDialogs
```

# PathItems

The collection of `pathItem` objects in the document. See [PathItem](#) for information on `pathItem` objects.

**Note:** Because the `PathItems` class is a property of the [Document](#) object, you use the property name, `pathItems`, rather than the class name, `PathItems`, in your code. For example:

```
var myPathItem = docRef.pathItems.add("A Line", lineSubPathArray)
```

The following sample uses the `PathItems` object incorrectly:

```
var myPathItem = docRef.PathItems.add("A Line", lineSubPathArray)
```

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of <code>pathItem</code> objects in the <code>pathItems</code> collection.
<b>parent</b>	object (document)	Read-only. The <code>pathItems</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>pathItems</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">PathItem</a>	Gets a <code>pathItem</code> object from the <code>pathItems</code> collection.
<b>add</b> (name, entirePath)	string array of <a href="#">SubPathInfo</a> objects	<a href="#">PathItem</a>	Creates a new <code>PathItem</code> object from the sub paths defined in the array provided in the <code>entirePath</code> parameter.  A new <a href="#">SubPathItem</a> object is created for each <code>SubPathInfo</code> object provided in <code>entirePath</code> , and those <code>SubPathItem</code> objects are added to the <a href="#">SubPathItems</a> collection of the returned <code>PathItem</code> .
<b>getByName</b> (name)	string	<a href="#">PathItem</a>	Get the first element in the <code>pathItems</code> collection with the provided name.
<b>removeAll</b> ( )			Removes all <code>pathItem</code> objects from the <code>pathItems</code> collection.

## PathPoint

Information about an array of `PathPointInfo` objects.

**Note:** You do not use the `PathPoint` object to create points that make up a path. Rather, you use the `PathPoint` object to retrieve information about the points that describe path segments. To create path points, use the `PathPointInfo` objects. See [PathPointInfo](#).

### Properties

Property	Value Type	What it is
<code>anchor</code>	array (UnitValue)	Read-only. The point on the curve ( <code>leftDirection</code> / <code>rightDirection</code> are points representing the control handle end points).  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<code>kind</code>	<a href="#">PointKind</a>	Read-only. The <code>PathPoint</code> object's type.
<code>leftDirection</code>	array (UnitValue)	Read-only. The x and y coordinates that define the left handle.  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<code>parent</code>	object ( <a href="#">SubPathItem</a> )	Read-only. The <code>PathPoint</code> object's container.
<code>rightDirection</code>	array (UnitValue)	Read-only. The x and y coordinates that define the right handle.  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<code>typename</code>	string	Read-only. The class name of the referenced <code>PathPoint</code> object.



## PathPointInfo

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

### Properties

Property	Value Type	What it is
<b>anchor</b>	array	Read-write. The x and y coordinates of one end point of the path segment.
<b>kind</b>	<a href="#">PointKind</a>	Read-write. The <code>PathPointInfo</code> object's kind.
<b>leftDirection</b>	array of <code>UnitValue</code>	Read-write. The location of the left direction point ('in' position).  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<b>rightDirection</b>	array of <code>UnitValue</code>	Read-write. The location of the right handle ('out' position).  <b>Note:</b> For information about the <code>UnitValue</code> type, see the <i>JavaScript Tools Guide</i> .
<b>typename</b>	string	Read-only. The class name of the referenced <code>PathPointInfo</code> object.

## PathPoints

A collection of `PathPoint` objects that comprises the `PathPoints` property of the `SubPathItem` object. See [SubPathItem](#) for more information.

### Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>PathPoints</code> collection.
<b>parent</b>	object ( <a href="#">SubPathItem</a> )	Read-only. The <code>PathPoints</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>PathPoints</code> object.

### Methods

Method	Parameter type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">PathPoint</a>	Gets an element from the <code>PathPoints</code> collection.

# PDFOpenOptions

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

## Properties

Property	Value Type	What it is
<b>antiAlias</b>	boolean	Read-write. Indicates whether to use antialias.
<b>bitsPerChannel</b>	<a href="#">BitsPerChannelType</a>	Read-write. The number of bits per channel.
<b>constrainProportions</b>	boolean	Deprecated for Adobe Photoshop CS3.
<b>cropPage</b>	<a href="#">CropToType</a>	Read-write. The method of cropping to use.
<b>height</b>	UnitValue	Deprecated for Adobe Photoshop CS3.
<b>mode</b>	<a href="#">OpenDocumentMode</a>	Read-write. The color model to use.
<b>name</b>	string	Read-write. The name of the document.
<b>page</b>	number (long)	Read-write. The page to which to open the document.
<b>resolution</b>	number (double)	Read-write. The resolution of the document (in pixels per inch).
<b>suppressWarnings</b>	boolean	Read-write. Indicates whether to suppress warnings when opening the document.
<b>typename</b>	string	Read-only. The class name of the referenced <code>PDFOpenOptions</code> object.
<b>usePageNumber</b>	boolean	Read-write. Indicates whether the value specified in the <code>page</code> property will refer to an image number when <code>usePageNumber = false</code> . See <a href="#">page</a> .
<b>width</b>	UnitValue	Deprecated for Adobe Photoshop CS3.

# PDFSaveOptions

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

## Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels with the file.
<b>annotations</b>	boolean	Read-write. Indicates whether to save comments with the file.
<b>colorConversion</b>	boolean	Read-write. Indicates whether to convert the color profile to a destination profile.
<b>convertToEightBit</b>	boolean	Read-write. Indicates whether to convert a 16-bit image to 8-bit for better compatibility with other applications.
<b>description</b>	string	Read-write. Description of the save options to use.
<b>destinationProfile</b>	string	Read-write. Description of the final RGB or CMYK output device, such as a monitor or a press standard.
<b>downgradeColorProfile</b>	boolean	Deprecated for Adobe Photoshop CS3.
<b>downSample</b>	<a href="#">PDFResample</a>	Read-write. The down sample method to use.
<b>downSampleSize</b>	number (double)	Read-write. The size to downsample images if they exceed the limit in pixels per inch.
<b>downSampleSizeLimit</b>	number (double)	Read-write. Limits downsampling or subsampling to images that exceed this value in pixels per inch.
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<b>embedFonts</b>	boolean	Deprecated for Adobe Photoshop CS3.
<b>embedThumbnail</b>	boolean	Read-write. Indicates whether to include a small preview image in Adobe PDF files.
<b>encoding</b>	<a href="#">PDFEncoding</a>	Read-write. The encoding method to use (default: <code>PDFEncoding.PDFZIP</code> ).
<b>interpolation</b>	boolean	Deprecated for Adobe Photoshop CS3.

Property	Value Type	What it is (Continued)
<code>jpegQuality</code>	number (long)	Read-write. The quality of the produced image (0 - 12), which is inversely proportionate to the compression amount.  <b>Note:</b> Valid only when <code>encoding = PDFEncoding.JPEG</code> .
<code>layers</code>	boolean	Read-write. Indicates whether to save the document's layers.
<code>optimizeForWeb</code>	boolean	Read-write. Indicates whether to improve performance of PDF files on Web servers.
<code>outputCondition</code>	string	Read-write. An optional comment field for inserting descriptions of the output condition. The text is stored in the PDF/X file.
<code>outputConditionID</code>	string	Read-write. Identifier for the output condition.
<code>PDFCompatibility</code>	<a href="#">PDFCompatibility</a>	Read-write. The PDF version to make the document compatible with.
<code>PDFStandard</code>	<a href="#">PDFStandard</a>	Read-write. The PDF standard to make the document compatible with.
<code>preserveEditing</code>	boolean	Read-write. Indicates whether to reopen the PDF in Adobe Photoshop CS3 with native Photoshop data intact.
<code>presetFile</code>	string	Read-write. The preset file to use for settings.  <b>Note:</b> This option overrides other settings.
<code>profileInclusionPolicy</code>	boolean	Read-write. Indicates whether to show which profiles to include.
<code>registryName</code>	string	Read-write. URL where the output condition is registered.
<code>spotColors</code>	boolean	Read-write. Indicates whether to save spot colors.
<code>tileSize</code>	number (long)	Read-write. Compression option.  <b>Note:</b> Valid only when <code>encoding = PDFEncoding.JPEG2000</code> .
<code>transparency</code>	boolean	Deprecated for Adobe Photoshop CS3.
<code>typename</code>	string	Read-only. The class name of the referenced <code>PDFSaveOptions</code> object.
<code>useOutlines</code>	boolean	Deprecated for Adobe Photoshop CS3.

Property	Value Type	What it is (Continued)
<b>vectorData</b>	boolean	Deprecated for Adobe Photoshop CS3.
<b>view</b>	boolean	Read-write. Indicates whether to open the saved PDF in Adobe Acrobat.

## PhotoCDOpenOptions

**Deprecated** in Adobe Photoshop CS3. Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS3 Install DVD.

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

### Properties

Property	Value Type	What it is
<code>colorProfileName</code>	string	Read-write. The profile to use when reading the image.
<code>colorSpace</code>	<a href="#">PhotoCDColorSpace</a>	Read-write. The colorspace for the image.
<code>orientation</code>	<a href="#">Orientation</a>	Read-write. The image orientation.
<code>pixelSize</code>	<a href="#">PhotoCDSize</a>	Read-write. The image dimensions.
<code>resolution</code>	number (double)	Read-write. The image resolution (in pixels per inch).
<code>typename</code>	string	Read-only. The class name of the referenced <code>photoCDOpenOptions</code> object.

# PhotoshopSaveOptions

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

## Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels.
<b>annotations</b>	boolean	Read-write. Indicates whether to save the annotations.
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<b>layers</b>	boolean	Read-write. Indicates whether to preserve the layers.
<b>spotColors</b>	boolean	Read-write. Indicates whether to save the spot colors.
<b>typename</b>	string	Read-only. The class name of the referenced <code>photoshopSaveOptions</code> object.



# PICTFileSaveOptions

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

## Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels.
<b>compression</b>	<a href="#">PICTCompression</a>	Read-write. (default: <code>PICTCompression.NONE</code> )
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<b>resolution</b>	<a href="#">PICTBitsPerPixels</a>	Read-write. The number of bits per pixel.
<b>typename</b>	string	Read-only. The class name of the referenced <code>PICTFileSaveOptions</code> object.

## PICTResourceSaveOptions

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

### Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels.
<b>compression</b>	<a href="#">PICTCompression</a>	Read-write. The type of compression to use (default: <code>PICTCompression.NONE</code> ).
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<b>name</b>	string	Read-write. The name of the PICT resource.
<b>resolution</b>	<a href="#">PICTBitsPerPixels</a>	Read-write. The number of bits per pixel.
<b>resourceID</b>	number (long)	Read-write. The ID of the PICT resource (default: 128).
<b>typename</b>	string	Read-only. The class name of the referenced <code>PICTResourceSaveOptions</code> object.

# PicturePackageOptions

Options that can be specified for a Picture Package.

## Properties

Property	Value type	What it is
<b>content</b>	<a href="#">PicturePackageTextType</a>	Read-write. The content information (default: <code>PicturePackageTextType.NONE</code> ).
<b>flatten</b>	boolean	Read-write. Indicates whether all layers in the final document are flattened (default: <code>true</code> ).
<b>font</b>	<a href="#">GalleryFontType</a>	Read-write. The font used for security text (default: <code>GalleryFontType.ARIAL</code> ).
<b>fontSize</b>	number (long)	Read-write. The font size used for security text (default: <code>12</code> ).
<b>layout</b>	string	Read-write. The layout to use to generate the picture package (default: <code>"(2) 5x7"</code> ).
<b>mode</b>	<a href="#">NewDocumentMode</a>	Read-write. Read-write. The color profile to use as the document mode (default: <code>NewDocumentMode.RGB</code> ).
<b>opacity</b>	number (long)	Read-write. The web page security opacity as a percent (default: <code>100</code> ).
<b>resolution</b>	number (double)	Read-write. The resolution of the document in pixels per inch (default: <code>72.0</code> ).
<b>text</b>	string	Read-write. The picture package custom text.  <b>Note:</b> Valid only when <code>content = PicturePackageType.USER</code> . See <a href="#">content</a> .
<b>textColor</b>	<a href="#">RGBColor</a>	Read-write. The color to use for security text.
<b>textPosition</b>	<a href="#">GallerySecurityTextPositionType</a>	Read-write. The security text position (default: <code>GallerySecurityTextPositionType.CENTERED</code> ).
<b>textRotate</b>	<a href="#">GallerySecurityTextRotateType</a>	Read-write. The orientation to use for security text (default: <code>GallerySecurityTextRotateType.ZERO</code> ).
<b>typename</b>	string	Read-only. The class name of the referenced <code>PicturePackageOptions</code> object.

## PixarSaveOptions

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

### Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels.
<b>typename</b>	string	Read-only. The class name of the referenced <code>PixarSaveOptions</code> object.

# PNGSaveOptions

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

## Properties

Property	Value Type	What it is
<b>interlaced</b>	boolean	Read-write. Indicates whether the should rows be interlaced (default: <code>false</code> ).
<b>typename</b>	string	Read-only. The class name of the referenced <code>PNGSaveOptions</code> object.

## Preferences

Options to define for the `preferences` property of the `app` object. See [preferencesFolder](#) (in the Properties table for the `app` object).

**Note:** Because the `Preferences` class is a property of the [Application](#) object, you use the property name, `preferences`, rather than the class name, `Preferences`, in your code. For example:

```
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
```

The following code incorrectly uses an upper case *P*:

```
app.Preferences.rulerUnits = Units.PIXELS
app.Preferences.typeUnits = TypeUnits.PIXELS
```

**Note:** Defining the `preferences` properties is basically equivalent to selecting Edit > Preferences (Windows) or Photoshop > Preferences in the Adobe Photoshop CS3 application. For explanations of individual settings, please refer to Adobe Photoshop CS3 Help.

## Properties

Property	Value Type	What it is
<code>additionalPluginFolder</code>	File	Read-write. The path to an additional plug-in folder.  <b>Note:</b> Valid only when <code>useAdditionalPluginFolder = true</code> . See <a href="#">useAdditionalPluginFolder</a> .  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i> .
<code>appendExtension</code>	<a href="#">SaveBehavior</a>	Read-write. Save files with extensions on Windows.
<code>askBeforeSavingLayeredTIFF</code>	boolean	Read-write. Indicates whether to ask the user to verify layer preservation options when saving a file in TIFF format.
<code>autoUpdateOpenDocuments</code>	boolean	Read-write. Indicates whether to automatically update open documents.
<code>beepWhenDone</code>	boolean	Read-write. Indicates whether to beep when a process finishes.
<code>colorChannelsInColor</code>	boolean	Read-write. Indicates whether to display component channels in the Channels palette in color.
<code>colorPicker</code>	<a href="#">ColorPicker</a>	Read-write.
<code>columnGutter</code>	number (double)	Read-write. The width of the column gutters (in points). (0.1 - 600.0).

Property	Value Type	What it is (Continued)
<code>columnWidth</code>	number (double)	Read-write. Column width (in points) (0.1 - 600.0).
<code>createFirstSnapshot</code>	boolean	Read-write. Indicates whether to automatically make the first snapshot when a new document is created.
<code>dynamicColorSliders</code>	boolean	Read-write. Indicates whether dynamic color sliders appear in the Color palette.
<code>editLogItems</code>	<a href="#">EditLogItemsType</a>	Read-write. The options for editing history log items.  <b>Note:</b> Valid only when <code>useHistoryLog = true</code> . See <a href="#">useHistoryLog</a> .
<code>exportClipboard</code>	boolean	Read-write. Indicates whether to retain Adobe Photoshop CS3 contents on the clipboard after you exit the application.
<code>fontPreviewSize</code>	<a href="#">FontPreviewType</a>	Read-write. Indicates whether to show font previews in the type tool font menus.
<code>fullSizePreview</code>	boolean	Read-write. (Mac only.) Indicates whether to show image preview as a full size image or thumbnail.
<code>gamutWarningOpacity</code>	number (double)	Read-write. (0 - 100 as percent).
<code>gridSize</code>	<a href="#">GridSize</a>	Read-write. The size to use for squares in the grid.
<code>gridStyle</code>	<a href="#">GridLineStyle</a>	Read-write. The formatting style for non-printing grid lines.
<code>gridSubDivisions</code>	number (long)	Read-write. (1 - 100)
<code>guideStyle</code>	<a href="#">GuideLineStyle</a>	Read-write. The formatting style for non-printing guide lines.
<code>iconPreview</code>	boolean	Read-write. (Mac only.)
<code>imageCacheLevels</code>	number (long)	Read-write. The number of images to hold in the cache (1 - 8).
<code>imagePreviews</code>	<a href="#">SaveBehavior</a>	Read-write. The behavior mode to use when saving files.
<code>interpolation</code>	<a href="#">ResampleMethod</a>	Read-write. The method to use to assign color values to any new pixels created when an image is resampled or resized.

Property	Value Type	What it is (Continued)
<code>keyboardZoomResizesWindows</code>	boolean	Read-write. Indicates whether to automatically resize the window when zooming in or out using keyboard shortcuts.
<code>macOSThumbnail</code>	boolean	Read-write. (Mac only.) Indicates whether to create a thumbnail when saving the image.
<code>maximizeCompatibility</code>	<a href="#">QueryStateType</a>	Read-write. The behavior to use to check whether to maximize compatibility when opening Adobe Photoshop CS3 (PSD) files .
<code>maxRAMuse</code>	number (long)	Read-write. The maximum percentage of available RAM used by Adobe Photoshop CS3 (5 - 100).
<code>nonLinearHistory</code>	boolean	Read-write. Indicates whether to allow non-linear history.
<code>numberOfHistoryStates</code>	number (long)	Read-write. The number of history states to preserve (1 - 100).
<code>otherCursors</code>	<a href="#">OtherPaintingCursors</a>	Read-write. The type of pointer to use.
<code>paintingCursors</code>	<a href="#">PaintingCursors</a>	Read-write. The type of pointer to use.
<code>parent</code>	object ( <a href="#">Application</a> )	Read-write. The <code>preferences</code> object's container.
<code>pixelDoubling</code>	boolean	Read-write. Indicates whether to halve the resolution or (double the size of pixels) to make previews display more quickly.
<code>pointSize</code>	<a href="#">PointType</a>	Read-write. The point/pica size.
<code>recentFileListLength</code>	number (long)	Read-write. The number of items in the recent file list (0 - 30).
<code>rulerUnits</code>	<a href="#">Units</a>	Read-write. The unit the scripting system will use when receiving and returning values.
<code>saveLogItems</code>	<a href="#">SaveLogItemsType</a>	Read-write. The options for saving the history items.
<code>saveLogItemsFile</code>	File	Read-write. The path to the history log file.  <b>Note:</b> For information about the File object, see the <i>JavaScript Tools Guide</i>



Property	Value Type	What it is (Continued)
<b>savePaletteLocations</b>	boolean	Read-write. Indicates whether to make new palette locations the default location.
<b>showAsianTextOptions</b>	boolean	Read-write. Indicates whether to display Asian text options in the Paragraph palette.
<b>showEnglishFontNames</b>	boolean	Read-write. Indicates whether to list Asian font names in English.
<b>showSliceNumber</b>	boolean	Read-write. Indicates whether to display slice numbers in the document window when using the Slice tool.
<b>showToolTips</b>	boolean	Read-write. Indicates whether to show pop up definitions on mouse over.
<b>smartQuotes</b>	boolean	Read-write. Indicates whether to use curly or straight quote marks.
<b>typename</b>	string	Read-only. The class name of the referenced <code>preferences</code> object.
<b>typeUnits</b>	<a href="#">TypeUnits</a>	Read-write. The unit type-size that the numeric inputs are assumed to represent.
<b>useAdditionalPluginFolder</b>	boolean	Read-write. Indicates whether to use an additional folder for compatible plug-ins stored with a different application.
<b>useHistoryLog</b>	boolean	Read-write. Indicates whether to create a log file for history states.
<b>useLowerCaseExtension</b>	boolean	Read-write. Indicates whether the file extension should be lowercase.
<b>useShiftKeyForToolSwitch</b>	boolean	Read-write. Indicates whether to enable cycling through a set of hidden tools.
<b>useVideoAlpha</b>	boolean	Read-write. Indicates whether to enable Adobe Photoshop CS3 to send transparency information to your computer's video board. (Requires hardware support.)
<b>windowsThumbnail</b>	boolean	Read-write. (Requires hardware support.) Indicates whether to create a thumbnail when saving the image on Windows.

# PresentationOptions

Options that can be specified for Adobe PDF presentations.

## Properties

Property	Value Type	What it is
<b>autoAdvance</b>	boolean	Read-write. Indicates whether to auto advance images when viewing the presentation (default: <code>true</code> ).  <b>Note:</b> Valid only when <code>presentation = true</code> . See <a href="#">presentation</a> .
<b>includeFilename</b>	boolean	Read-write. Indicates whether to include the file name for the image (default: <code>false</code> ).
<b>interval</b>	number (long)	Read-write. The time in seconds before the view is auto advanced (1 - 60; default: 5).  <b>Note:</b> Valid only when <code>AutoAdvance = true</code> . See <a href="#">autoAdvance</a> .
<b>loop</b>	boolean	Read-write. Indicates whether to begin the presentation again after the last page (default: <code>false</code> ).  <b>Note:</b> Valid only when <code>autoAdvance = true</code> . See <a href="#">autoAdvance</a> .
<b>magnification</b>	<a href="#">MagnificationType</a>	Read-write. The magnification type to use when viewing the image.
<b>pdfFileOptions</b>	<a href="#">PDFSaveOptions</a>	Read-write. Options to use when creating the PDF file.
<b>presentation</b>	boolean	Read-write. Indicates whether the output will be a presentation (default: <code>false</code> ); when <code>false</code> , the output is a Multi-Page document.
<b>transition</b>	<a href="#">TransitionType</a>	Read-write. The transition from one image to the next (default: <code>TransitionType.NONE</code> ).  <b>Note:</b> Valid only when <code>autoAdvance = true</code> . See <a href="#">autoAdvance</a> .
<b>typename</b>	string	Read-only. The class name of the referenced <code>PresentationOptions</code> object.

# RawFormatOpenOptions

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

## Properties

Property	Value Type	What it is
<b>bitsPerChannel</b>	number (long)	Read-write. The number of bits for each channel.  <b>Note:</b> The only valid values are <code>bitsPerChannel = BitsPerChannelType.EIGHT</code> or <code>bitsPerChannel = BitsPerChannelType.SIXTEEN</code> .
<b>byteOrder</b>	<a href="#">ByteOrder</a>	Read-write. The order in which bytes will be read.  <b>Note:</b> Valid only when <code>bitsPerChannel = BitsPerChannelType.SIXTEEN</code> . See <a href="#">bitsPerChannel</a> .
<b>channelNumber</b>	number (long)	Read-write. The number of channels in the image (1 - 56).  <b>Note:</b> The value of <code>channelNumber</code> cannot exceed the number of channels in the image. When <code>bitsPerChannel = BitsPerChannelType.SIXTEEN</code> , only the following values are valid: 1, 3, or 4. See <a href="#">bitsPerChannel</a> .
<b>headerSize</b>	number (long)	Read-write. The number of bytes of information that will appear in the file before actual image information begins; that is, the number of zeroes inserted at the beginning of the file as placeholders (0 - 1919999).
<b>height</b>	number (long)	Read-write. The height of the image (in pixels).
<b>interleaveChannels</b>	boolean	Read-write. Indicates whether to store color values sequentially.
<b>retainHeader</b>	boolean	Read-write. Indicates whether to retain the header when saving.  <b>Note:</b> Valid only when <a href="#">headerSize</a> is 1 or greater.
<b>typename</b>	string	Read-only. The class name of the referenced <code>RawFormatOpenOptions</code> object.
<b>width</b>	number (long)	Read-write. The image width in pixels.

## RawSaveOptions

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

### Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether alpha channels should be saved.
<b>spotColors</b>	boolean	Read-write. Indicates whether the spot colors should be saved.
<b>typename</b>	string	Read-only. The class name of the referenced <code>RawSaveOptions</code> object.

# RGBColor

The definition of a color in RGB color mode.

## Properties

Property	Value Type	What it is
<b>blue</b>	number (double)	Read-write. The blue color value (0.0 - 255.0; default: 255.0).
<b>green</b>	number (double)	Read-write. The green color value (0.0 - 255.0; default: 255.0).
<b>hexValue</b>	string	Read-write. The hex representation of the color.
<b>red</b>	number (double)	Read-write. The red color value (0.0 - 255.0; default: 255.0).
<b>typename</b>	string	Read-only. The class name of the referenced <code>RGBColor</code> object.

## Selection

The selected area of a document or layer.

**Note:** Many of the properties and methods of `Selection` use the `UnitValue` type. For information about this type, see the *JavaScript Tools Guide*.

**Note:** Because the `Selection` class is a property of the [Document](#) object, you use the property name, `selection`, rather than the class name, `Selection`, in your code, as in the following example:

```
checkersDoc.selection.fill(app.foregroundColor)
```

## Properties

Property	Value Type	What it is
<b>bounds</b>	array of <code>UnitValue</code>	Read-only. The bounding rectangle of the entire selection.
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The object's container.
<b>solid</b>	boolean	Read-only. Indicates if the bounding rectangle is a solid.
<b>typename</b>	string	Read-only. The class name of the referenced <code>selection</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>clear</b> ( )			Clears the selection and does not copy it to the clipboard.
<b>contract</b> (by)	<code>UnitValue</code>		Contracts the selection by the specified amount.
<b>copy</b> ( [merge] )	boolean		Copies the selection to the clipboard. When the optional argument is used and set to <code>true</code> , a merged copy is performed (all visible layers in the selection are copied).
<b>cut</b> ( )			Clears the current selection and copies it to the clipboard.
<b>deselect</b> ( )			Deselects the current selection.
<b>expand</b> (by)	<code>UnitValue</code>		Expands the selection by the specified amount.
<b>feather</b> (by)	<code>UnitValue</code>		Feathers the edges of the selection by the specified amount.

Method	Parameter Type	Returns	What it does (Continued)
<b>fill</b> (filltype [, mode] [, opacity] [, preserveTransparency])	Object ( <a href="#">SolidColor</a> or <a href="#">HistoryState</a> ) <a href="#">ColorBlendMode</a> number (long) boolean		Fills the selection (opacity: 1 - 100 as percent).
<b>grow</b> (tolerance, antiAlias)	number (long) boolean		Grows the selection to include all adjacent pixels falling within the specified tolerance range.
<b>invert</b> ( )			Inverts the selection (deselects the selection and selects the rest of the layer or document).  <b>Note:</b> To flip the selection shape, see <a href="#">rotate</a> .
<b>load</b> (from [, combination] [, inverting])	<a href="#">Channel</a> <a href="#">SelectionType</a> boolean		Loads the selection from the specified channel.
<b>makeWorkPath</b> ([tolerance])	number (double)		Makes this selection item the work path for this document.
<b>resize</b> ([horizontal] [, vertical] [, anchor])	number (double) number (double) <a href="#">AnchorPosition</a>		Resizes the selected area to the specified dimensions and anchor position.
<b>resizeBoundary</b> ([horizontal] [, vertical] [, anchor])	number (double) number (double) <a href="#">AnchorPosition</a>		Changes the size of the selection to the specified dimensions around the specified anchor.
<b>rotate</b> (angle [, anchor])	number (double) <a href="#">AnchorPosition</a>		Rotates the selection by the specified amount around the specified anchor point.
<b>rotateBoundary</b> (angle [, anchor])	number (double) <a href="#">AnchorPosition</a>		Rotates the boundary of the selection around the specified anchor.
<b>select</b> (region [, type] [, feather] [, antiAlias])	array of points: Array (Array (x,y),...) <a href="#">SelectionType</a> number (double) boolean		Selects the specified region.
<b>selectAll</b> ( )			Selects the entire layer.

Method	Parameter Type	Returns	What it does (Continued)
<b>selectBorder</b> (width)	UnitValue		Selects the selection border only (in the specified width); subsequent actions do not affect the selected area within the borders.
<b>similar</b> (tolerance, antiAlias)	number (long) boolean		Grows the selection to include pixels throughout the image falling within the tolerance range.
<b>smooth</b> (radius)	number (long)		Cleans up stray pixels left inside or outside a color-based selection (within the radius specified in pixels).
<b>store</b> (into [, combination])	<a href="#">Channel</a> <a href="#">SelectionType</a>		Saves the selection as a channel.
<b>stroke</b> (strokeColor, width [, location] [, mode] [, opacity] [, preserveTransparency])	Object ( <a href="#">SolidColor</a> ) number (long) <a href="#">StrokeLocation</a> <a href="#">ColorBlendMode</a> number (long) boolean		Strokes the selection border (opacity: 1 - 100 as percent).
<b>translate</b> ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the entire selection relative to its current position.
<b>translateBoundary</b> ([deltaX] [, deltaY])	UnitValue UnitValue		Moves the selection relative to its current position.

## Sample Script

The following script creates a checkerboard using the following steps:

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

### Selection.jsx

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits
var startTypeUnits = app.preferences.typeUnits
```



```
var startDisplayDialogs = app.displayDialogs

// Set Adobe Photoshop CS3 to use pixels and display no dialogs
app.preferences.rulerUnits = Units.PIXELS
app.preferences.typeUnits = TypeUnits.PIXELS
app.displayDialogs = DialogModes.NO

//Close all the open documents
while (app.documents.length) {
    app.activeDocument.close()
}

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

// create a new document
var checkersDoc = app.documents.add(docSize, docSize, 72, "Checkers")

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

// loop through vertically to create the first row
for (var v = 0; v < docSize; v += cellSize) {

    // Switch the shift for a new row
    shiftIt = !shiftIt

    // loop through horizontally
    for (var h = 0; h < docSize; h += (cellSize * 2)) {

        // push over the cellSize to start with only
        if (shiftIt && h == 0) {
            h += cellSize
        }

        // Select a square
        selRegion = Array(Array(h, v),
            Array(h + cellSize, v),
            Array(h + cellSize, v + cellSize),
            Array(h, v + cellSize),
            Array(h, v))

        // In the first iteration of the loop, start the selection
        //In subsequent iterations, use the EXTEND constant value
        //of the select() method to add to the selection (in the loop's else clause)
        if (h == 0 && v == 0) {
            checkersDoc.selection.select(selRegion)
        } else {
            checkersDoc.selection.select(selRegion, SelectionType.EXTEND)
        }

        // turn this off for faster execution
        // turn this on for debugging
        WaitForRedraw()
    }
}
```

```
// Fill the current selection with the foreground color
checkersDoc.selection.fill(app.foregroundColor)

//Invert the selection
checkersDoc.selection.invert()

// Fill the new selection with the background color
checkersDoc.selection.fill(app.backgroundColor)

// Clear the selection to get rid of the non-printing borders
checkersDoc.selection.deselect()

// Reset the application preferences
app.preferences.rulerUnits = startRulerUnits
app.preferences.typeUnits = startTypeUnits
app.displayDialogs = startDisplayDialogs

// A helper function for debugging
// It also helps the user see what is going on
// if you turn it off for this example you
// get a flashing cursor for a number (long) time
function WaitForRedraw()
{
    var eventWait = charIDToTypeID("Wait")
    var enumRedrawComplete = charIDToTypeID("RdCm")
    var typeState = charIDToTypeID("Stte")
    var keyState = charIDToTypeID("Stte")

    var desc = new ActionDescriptor()

    desc.putEnumerated(keyState, typeState, enumRedrawComplete)

    executeAction(eventWait, desc, DialogModes.NO)
}
```

## SGIRGBSaveOptions

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

**Note:** The SGIRGB format is not installed automatically with Adobe Photoshop CS3.

### Properties

Property	Value Type	What it is
<code>alphaChannels</code>	boolean	Read-write. Indicates whether to save the alpha channels.
<code>spotColors</code>	boolean	Read-write. Indicates whether to save the spot colors.
<code>typename</code>	string	Read-only. The class name of the referenced <code>SGIRGBSaveOptions</code> object.

# SolidColor

A color definition used in the document.

## Properties

Property	Value Type	What it is
<b>cmyk</b>	<a href="#">CMYKColor</a>	Read-write. The CMYK color mode.
<b>gray</b>	<a href="#">GrayColor</a>	Read-write. The Grayscale color mode.
<b>hsb</b>	<a href="#">HSBColor</a>	Read-write. The HSB color mode.
<b>lab</b>	<a href="#">LabColor</a>	Read-write. The LAB color mode.
<b>model</b>	<a href="#">ColorModel</a>	Read-write. The color model.
<b>nearestWebColor</b>	<a href="#">RGBColor</a>	Read-only. The nearest web color to the current color.
<b>rgb</b>	<a href="#">RGBColor</a>	Read-write. The RGB color mode.
<b>typename</b>	string	Read-only. The class name of the referenced <code>SolidColor</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>isEqual</b> (color)	<a href="#">SolidColor</a>	boolean	Indicates whether the <code>SolidColor</code> object is visually equal to the specified color.

## SubPathInfo

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

You add sub path information to a path by passing a `SubPathInfo` object into the [add](#) method of the [PathItems](#) class. This method creates the `SubPathItem` objects associated with each `SubPathInfo` object, and returns a `PathItem` object that is the path represented by all the sub paths.

### Properties

Property	Value Type	What it is
<b>closed</b>	boolean	Read-write. Indicates whether the path describes an enclosed area.
<b>entireSubPath</b>	Array ( <a href="#">PathPoint</a> objects)	Read-write.
<b>operation</b>	<a href="#">ShapeOperation</a>	Read-write. The sub path's operation on other sub paths.
<b>typename</b>	string	Read-only. The class name of the referenced <code>SubPathInfo</code> object.

## SubPathItem

Information about a path.

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

### Properties

Property	Value Type	What it is
<code>closed</code>	boolean	Read-only. Indicates whether the path is closed.
<code>operation</code>	<a href="#">ShapeOperation</a>	Read-only. The sub path operation on other sub paths.
<code>parent</code>	object ( <a href="#">PathItem</a> )	Read-only. The object's container.
<code>pathPoints</code>	<a href="#">PathPoints</a>	Read-only. The <code>PathPoints</code> collection.
<code>typename</code>	string	Read-only. The class name of the referenced <code>SubPathItem</code> object.

# SubPathItems

A collection of `SubPathItem` objects. See [SubPathItem](#).

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>SubPathItems</code> collection.
<b>parent</b>	object ( <a href="#">PathItem</a> )	Read-only. The <code>SubPathItems</code> object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>SubPathItems</code> object.

## Methods

Method	Parameter type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">SubPathItem</a>	Gets an element from the <code>SubPathItems</code> collection.

# TargaSaveOptions

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

## Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels.
<b>resolution</b>	<a href="#">TargaBitsPerPixels</a>	Read-write. The number of bits per pixel (default: <code>TargaBitsPerPixels.TWENTYFOUR</code> ).
<b>rleCompression</b>	boolean	Read-write. Indicates whether RLE compression should be used (default: <code>true</code> ).
<b>typename</b>	string	Read-only. The class name of the referenced <code>TargaSaveOptions</code> object.



# TextFont

Details about a font in the `TextFonts` collection. See [TextFonts](#) for more information on the `TextFonts` collection.

## Properties

Property	Value Type	What it is
<b>family</b>	string	Read-only. The font family.
<b>name</b>	string	Read-only. The name of the font.
<b>parent</b>	object ( <a href="#">Application</a> )	Read-only. The object's container.
<b>postScriptName</b>	string	Read-only. The PostScript name of the font.
<b>style</b>	string	Read-only. The font style.
<b>typename</b>	string	Read-only. The class name of the referenced <code>TextFont</code> object.

## TextFonts

The collection of fonts available on your computer.

**Note:** The `TextFonts` class corresponds to the `fonts` property of the `TextFonts` object. In a script, you use the property name `fonts`, rather than the class name `TextFonts`, to refer to a `TextFonts` object. The following example uses the `length` property to determine, and then display, the number of `TextFonts` installed on the machine.

- Correct:  

```
alert (app.fonts.length)
```
- Incorrect:  

```
alert (app.TextFonts.length)
```

See [Application](#), specifically the `fonts` property, for more information.

## Properties

Property	Value Type	What it is
<b>length</b>	number (long)	Read-only. The number of elements in the <code>TextFonts</code> collection.
<b>parent</b>	object ( <a href="#">Application</a> )	Read-only. The object's container.
<b>typename</b>	string	Read-only. The class name of the referenced <code>TextFonts</code> object.

## Methods

Method	Parameter Type	Returns	What it does
<b>index</b> (itemKey)	number	<a href="#">TextFont</a>	Gets an element from the <code>TextFonts</code> collection.
<b>getByName</b> (name)	string	<a href="#">TextFont</a>	Gets the first element in the <code>TextFonts</code> collection with the provided name.

## TextItem

The text in an `artLayer` object whose `kind` property is `LayerKind.TEXT`. See [ArtLayer](#), specifically the `kind` property, for more information.

**Note:** Many of the properties in the `TextItem` class use the `UnitValue` type. For information about this type, see the *JavaScript Tools Guide*.

**Note:** Because the `TextItem` class is a property of the `ArtLayer` class, you use the property name, `textItem`, rather than the class name, `TextItem`, in your code. For example:

```
myLayers[i].textItem.contents = "Layer in " + textArray[i] + " Set Inside "
```

## Properties

Property	Value Type	What it is
<code>alternateLigatures</code>	boolean	Read-write. Indicates whether to use alternate ligatures.  <b>Note:</b> Alternate ligatures are the same as Discretionary Ligatures. Please refer to Adobe Photoshop CS3 Help for more information.
<code>antiAliasMethod</code>	<a href="#">AntiAlias</a>	Read-write. The method of anti aliasing to use.
<code>autoKerning</code>	<a href="#">AutoKernType</a>	Read-write. The auto kerning option to use.
<code>autoLeadingAmount</code>	number (double)	Read-write. The percentage to use for auto (default) leading (0.01 - 5000.00 in points).  <b>Note:</b> Valid only when <code>useAutoLeading = true</code> . See <a href="#">useAutoLeading</a> .
<code>baselineShift</code>	UnitValue	Read-write. The unit value to use in the baseline offset of text.
<code>capitalization</code>	<a href="#">TextCase</a>	Read-write. The text case.
<code>color</code>	<a href="#">SolidColor</a>	Read-write. The text color.
<code>contents</code>	string	Read-write. The actual text in the layer.
<code>desiredGlyphScaling</code>	number (double)	Read-write. The desired amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled).  <b>Note:</b> Valid only when <div> justification =  Justification.CENTERJUSTIFIED;  Justification.FULLYJUSTIFIED;  Justification.LEFTJUSTIFIED; or  Justification.RIGHTJUSTIFIED. See <a href="#">justification</a>. The following values are also required: <a href="#">minimumGlyphScaling</a> and <a href="#">maximumGlyphScaling</a>. </div>

Property	Value Type	What it is (Continued)
<b>desiredLetterScaling</b>  <b>Note:</b> 'Letter Scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS3 application Justification dialog (Select Justification on the Paragraphs palette menu).	number (double)	Read-write. The amount of space between letters (100 - 500; at 0, no space is added between letters).  <b>Note:</b> Valid only when <code>justification =</code> <code>Justification.CENTERJUSTIFIED;</code> <code>Justification.FULLYJUSTIFIED;</code> <code>Justification.LEFTJUSTIFIED;</code> or <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a> . The following values are also required: <a href="#">minimumLetterScaling</a> and <a href="#">maximumLetterScaling</a> .
<b>desiredWordScaling</b>  <b>Note:</b> 'Word Scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS3 application Justification dialog (Select Justification on the Paragraphs palette menu).	number (double)	Read-write. The amount (percentage) of space between words (0 -1000; at 100, no additional space is added between words).  <b>Note:</b> Valid only when <code>justification =</code> <code>Justification.CENTERJUSTIFIED;</code> <code>Justification.FULLYJUSTIFIED;</code> <code>Justification.LEFTJUSTIFIED;</code> or <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a> . The following values are also required: <a href="#">minimumWordScaling</a> and <a href="#">maximumWordScaling</a> .
<b>direction</b>	<a href="#">Direction</a>	Read-write. The text orientation.
<b>fauxBold</b>	boolean	Read-write. Indicates whether to use faux bold (default: <code>false</code> ).  <b>Note:</b> Using <code>fauxBold.true</code> is equivalent to selecting text and clicking the Faux Bold button in the Character palette.
<b>fauxItalic</b>	boolean	Read-write. Indicates whether to use faux italic (default: <code>false</code> ).  <b>Note:</b> Using <code>fauxItalic.true</code> is equivalent to selecting text and clicking the Faux Italic button in the Character palette.
<b>firstLineIndent</b>	UnitValue	Read-write. The amount (unit value) to indent the first line of paragraphs (-1296 - 1296).
<b>font</b>	string	Read-write. The text face of the character.
<b>hangingPunctuation</b>	boolean	Read-write. Indicates whether to use roman Hanging Punctuation.

Property	Value Type	What it is (Continued)
<b>height</b>	UnitValue	Read-write. The height of the bounding box (unit value) for paragraph text.  <b>Note:</b> Valid only when <code>kind = TextType.PARAGRAPHTEXT</code> . See <a href="#">kind</a> .
<b>horizontalScale</b>	number (long)	Read-write. Character scaling (horizontal) in proportion to vertical scale (0 - 1000 in percent). See <a href="#">verticalScale</a> .
<b>hyphenateAfterFirst</b>	number (long)	Read-write. The number of letters after which hyphenation in word wrap is allowed (1 - 15).
<b>hyphenateBeforeLast</b>	number (long)	Read-write. The number of letters before which hyphenation in word wrap is allowed (1 - 15).
<b>hyphenateCapitalWords</b>	boolean	Read-write. Indicates whether to allow hyphenation in word wrap of capitalized words.
<b>hyphenateWordsLongerThan</b>	number (long)	Read-write. The minimum number of letters a word must have in order for hyphenation in word wrap to be allowed (2 - 25).
<b>hyphenation</b>	boolean	Read-write. Indicates whether to use hyphenation in word wrap.
<b>hyphenationZone</b>	UnitValue	Read-write. The distance at the end of a line that will cause a word to break in unjustified type (0 - 720 pica).
<b>hyphenLimit</b>	number (long)	Read-write. The maximum number of consecutive lines that can end with a hyphenated word.
<b>justification</b>	<a href="#">Justification</a>	Read-write. The paragraph justification.
<b>kind</b>	<a href="#">TextType</a>	Read-write. The text-wrap type.
<b>language</b>	<a href="#">Language</a>	Read-write. The language to use.
<b>leading</b>	UnitValue	Read-write. The leading amount (unit value).
<b>leftIndent</b>	UnitValue	Read-write. The amount (unit value) of space to indent text from the left (-1296 - 1296).
<b>ligatures</b>	boolean	Read-write. Indicates whether to use ligatures.

Property	Value Type	What it is (Continued)
<b>maximumGlyphScaling</b>	number (double)	<p>Read-write. The maximum amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled).</p> <p><b>Note:</b> Valid only when  <code>justification =</code>  <code>Justification.CENTERJUSTIFIED;</code>  <code>Justification.FULLYJUSTIFIED;</code>  <code>Justification.LEFTJUSTIFIED;</code> or  <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a>. The following values are also required: <a href="#">minimumGlyphScaling</a> and <a href="#">desiredGlyphScaling</a>.</p>
<b>maximumLetterScaling</b>  <b>Note:</b> 'Letter Scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS3 application Justification dialog (Select Justification on the Paragraphs palette menu).	number (double)	<p>Read-write. The maximum amount of space to allow between letters (100 - 500; at 0, no space is added between letters).</p> <p><b>Note:</b> Valid only when  <code>justification =</code>  <code>Justification.CENTERJUSTIFIED;</code>  <code>Justification.FULLYJUSTIFIED;</code>  <code>Justification.LEFTJUSTIFIED;</code> or  <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a>. The following values are also required: <a href="#">minimumLetterScaling</a> and <a href="#">desiredLetterScaling</a>.</p>
<b>maximumWordScaling</b>  <b>Note:</b> 'Word Scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS3 application Justification dialog (Select Justification on the Paragraphs palette menu).	number (double)	<p>Read-write. The maximum amount (percentage) of space to allow between words (0 -1000; at 100, no additional space is added between words).</p> <p><b>Note:</b> Valid only when  <code>justification =</code>  <code>Justification.CENTERJUSTIFIED;</code>  <code>Justification.FULLYJUSTIFIED;</code>  <code>Justification.LEFTJUSTIFIED;</code> or  <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a>. The following values are also required: <a href="#">minimumWordScaling</a> and <a href="#">desiredWordScaling</a>.</p>

Property	Value Type	What it is (Continued)
<b>minimumGlyphScaling</b>	number (double)	<p>Read-write. The minimum amount (percentage) to scale the horizontal size of the text letters (50 - 200; at 100, the width of characters is not scaled).</p> <p><b>Note:</b> Valid only when  <code>justification =</code>  <code>Justification.CENTERJUSTIFIED;</code>  <code>Justification.FULLYJUSTIFIED;</code>  <code>Justification.LEFTJUSTIFIED;</code> or  <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a>. The following values are also required: <a href="#">minimumGlyphScaling</a> and <a href="#">desiredGlyphScaling</a>.</p>
<b>minimumLetterScaling</b>  <b>Note:</b> 'Letter Scaling' is basically equivalent to 'Letter Spacing' in the Adobe Photoshop CS3 application Justification dialog (Select Justification on the Paragraphs palette menu).	number (double)	<p>Read-write. The minimum amount (percentage) of space between letters (100 - 500; at 0, no space is removed between letters).</p> <p><b>Note:</b> Valid only when  <code>justification =</code>  <code>Justification.CENTERJUSTIFIED;</code>  <code>Justification.FULLYJUSTIFIED;</code>  <code>Justification.LEFTJUSTIFIED;</code> or  <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a>. The following values are also required: <a href="#">maximumLetterScaling</a> and <a href="#">desiredLetterScaling</a>.</p>
<b>minimumWordScaling</b>  <b>Note:</b> 'Word Scaling' is basically equivalent to 'Word Spacing' in the Adobe Photoshop CS3 application Justification dialog (Select Justification on the Paragraphs palette menu).	number (double)	<p>Read-write. The minimum amount (percentage) of space between words (0 -1000; at 100, no space is removed between words).</p> <p><b>Note:</b> Valid only when  <code>justification =</code>  <code>Justification.CENTERJUSTIFIED;</code>  <code>Justification.FULLYJUSTIFIED;</code>  <code>Justification.LEFTJUSTIFIED;</code> or  <code>Justification.RIGHTJUSTIFIED.</code> See <a href="#">justification</a>. The following values are also required: <a href="#">maximumWordScaling</a> and <a href="#">desiredWordScaling</a>.</p>
<b>noBreak</b>	boolean	<p>Read-write. Indicates whether to allow words to break at the end of a line.</p> <p><b>Tip:</b> 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.</p>
<b>oldStyle</b>	boolean	Read-write. Indicates whether to use old style type.

Property	Value Type	What it is (Continued)
<b>parent</b>	object ( <a href="#">ArtLayer</a> )	Read-write. The <code>TextItem</code> object's container.
<b>position</b>	array ( <code>UnitValue</code> )	Read-write. The position of origin for the text. The array must contain two values (unit value).  <b>Tip:</b> 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.
<b>rightIndent</b>	<code>UnitValue</code>	Read-write. The amount of space (unit value) to indent text from the right (-1296 - 1296).
<b>size</b>	number (double)	Read-write. The font size in points.
<b>spaceAfter</b>	<code>UnitValue</code>	Read-write. The amount of space (unit value) to use after each paragraph (-1296 - 1296).
<b>spaceBefore</b>	<code>UnitValue</code>	Read-write. The amount of space (unit value) to use before each paragraph (-1296 - 1296).
<b>strikeThru</b>	<a href="#">StrikeThruType</a>	Read-write. The text strike through option to use.
<b>textComposer</b>	<a href="#">TextComposer</a>	Read-write. The composition method to use to evaluate line breaks and optimize the specified hyphenation and justification options.  <b>Note:</b> Valid only when <code>kind = TextType.PARAGRAPHTEXT</code> . See <a href="#">kind</a> .
<b>tracking</b>	number (double)	Read-write. The amount of uniform spacing between multiple characters (-1000 - 10000).  <b>Note:</b> 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.
<b>typename</b>	string	Read-only. The class name of the referenced <code>textItem</code> object.
<b>underline</b>	<a href="#">UnderlineType</a>	Read-write. The text underlining options.
<b>useAutoLeading</b>	boolean	Read-write. Indicates whether to use a font's built-in leading information.
<b>verticalScale</b>	number (long)	Read-write. Character scaling (vertical) in proportion to horizontal scale (0 - 1000 in percent). See <a href="#">horizontalScale</a> .
<b>warpBend</b>	number (double)	Read-write. The warp bend percentage (-100 - 100).
<b>warpDirection</b>	<a href="#">Direction</a>	Read-write. The warp direction.



Property	Value Type	What it is (Continued)
<b>warpHorizontalDistortion</b>	number (double)	Read-write. The horizontal distortion (as percentage) of the warp (-100 - 100).
<b>warpStyle</b>	<a href="#">WarpStyle</a>	Read-write. The style of warp to use.
<b>warpVerticalDistortion</b>	number (double)	Read-write. The vertical distortion (as percentage) of the warp (-100 - 100).
<b>width</b>	UnitValue	Read-write. The width of the bounding box (unit value) for paragraph text.  <b>Note:</b> Valid only when <code>kind = TextType.PARAGRAPHTEXT</code> . See <a href="#">kind</a> .

## Methods

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

# TiffSaveOptions

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

## Properties

Property	Value Type	What it is
<b>alphaChannels</b>	boolean	Read-write. Indicates whether to save the alpha channels.
<b>annotations</b>	boolean	Read-write. Indicates whether to save the annotations.
<b>byteOrder</b>	<a href="#">ByteOrder</a>	Read-write. The order in which the document's bytes will be read. (The default is <code>ByteOrder.MACOS</code> when running on MacOS and <code>ByteOrder.IBM</code> when running on a PC.)
<b>embedColorProfile</b>	boolean	Read-write. Indicates whether to embed the color profile in the document.
<b>imageCompression</b>	<a href="#">TIFFEncoding</a>	Read-write. The compression type (default: <code>TIFFEncoding.NONE</code> ).
<b>interleaveChannels</b>	boolean	Read-write. Indicates whether the channels in the image will be interleaved.
<b>jpegQuality</b>	number (long)	Read-write. The quality of the produced image (0 - 12), which is inversely proportionate to the amount of JPEG compression.  <b>Note:</b> Valid only when <code>imageCompression = TIFFEncoding.JPEG</code> .
<b>layerCompression</b>	<a href="#">LayerCompression</a>	Read-write. The method of compression to use when saving layers (as opposed to saving composite data).  <b>Note:</b> Valid only when <code>layers = true</code> . See <a href="#">layers</a>
<b>layers</b>	boolean	Read-write. Indicates whether to save the layers.
<b>saveImagePyramid</b>	boolean	Read-write. Indicates whether to preserve multi-resolution information (default: <code>false</code> ).
<b>spotColors</b>	boolean	Read-write. Indicates whether to save the spot colors.
<b>transparency</b>	boolean	Read-write. Indicates whether to save the transparency as an additional alpha channel when the file is opened in another application.
<b>typename</b>	string	Read-only. The class name of the referenced <code>TiffSaveOptions</code> object.

## xmpMetadata

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

### Properties

Property	Value Type	What it is
<b>parent</b>	object ( <a href="#">Document</a> )	Read-only. The object's container.
<b>rawData</b>	string	Read-write. The raw XML form of file information.
<b>typename</b>	string	Read-only. The class name of the referenced <code>xmpMetadata</code> object.

# 3

## Scripting Constants

This section lists and describes the enumerations defined for use with Adobe Photoshop CS3 JavaScript properties and methods.

Constant type	Values	What it means
<b>AdjustmentReference</b>	ABSOLUTE RELATIVE	Method to use for interpreting selective color adjustment specifications: ABSOLUTE = % of the whole; RELATIVE = % of the existing color amount.
<b>AnchorPosition</b>	BOTTOMCENTER BOTTOMLEFT BOTTOMRIGHT MIDDLECENTER MIDDLELEFT MIDDLERIGHT TOPCENTER TOPLEFT TOPRIGHT	The point on the object that does not move when the object is rotated or resized.
<b>AntiAlias</b>	CRISP NONE SHARP SMOOTH STRONG	Method to use to smooth edges by softening the color transition between edge pixels and background pixels.
<b>AutoKernType</b>	MANUAL METRICS OPTICAL	The type of kerning to use for characters.
<b>BatchDestinationType</b>	FOLDER NODESTINATION SAVEANDCLOSE	The destination, if any, for batch-processed files: FOLDER: Save modified versions of the files to a new location (leaving the originals unchanged); NODESTINATIONTYPE: Leave all files open; SAVEANDCLOSE: Save changes and close the files.
<b>BitmapConversionType</b>	CUSTOMPATTERN DIFFUSIONDITHER HALFTHRESHOLD HALFTONESCREEN PATTERNDITHER	Specifies the quality of an image you are converting to bitmap mode.
<b>BitmapHalfToneType</b>	CROSS DIAMOND ELLIPSE LINE ROUND SQUARE	Specifies the shape of the dots (ink deposits) in the halftone screen.
<b>BitsPerChannelType</b>	EIGHT ONE SIXTEEN THIRTYTWO	The number of bits per color channel.

Constant type	Values	What it means
<b>BlendMode</b>	COLORBLEND COLORBURN COLORDODGE DARKEN DIFFERENCE DISSOLVE EXCLUSION HARDLIGHT HARDMIX HUE LIGHTEN LINEARBURN LINEARDODGE LINEARLIGHT LUMINOSITY MULTIPLY NORMAL OVERLAY PASSTHROUGH PINLIGHT SATURATION SCREEN SOFTLIGHT VIVIDLIGHT	Controls how pixels in the image are blended.
<b>BMPDepthType</b>	BMP_A1R5G5B5 BMP_A4R4G4B4 BMP_A8R8G8B8 BMP_R5G6B5 BMP_R8G8B8 BMP_X1R5G5B5 BMP_X4R4G4B4 BMP_X8R8G8B8 EIGHT FOUR ONE SIXTEEN THIRTYTWO TWENTYFOUR	The number of bits per channel (also called pixel depth or color depth). The number selected indicates the exponent of 2. For example, a pixel with a bit-depth of <code>EIGHT</code> has $2^8$ , or 256, possible color values.
<b>ByteOrder</b>	IBM MACOS	The order in which bytes will be read.
<b>CameraRAWSettingsType</b>	CAMERA CUSTOM SELECTEDIMAGE	The default CameraRaw settings to use: the camera settings, custom settings, or the settings of the selected image.
<b>CameraRAWSize</b>	EXTRALARGE LARGE MAXIMUM MEDIUM MINIMUM SMALL	The camera RAW size type options: EXTRALARGE=5120 x 4096 LARGE=4096 x 2731 MAXIMUM=6144 X 4096 MEDIUM=3072 x 2048 MINIMUM=1536 x 1024 SMALL=2048 x 1365

Constant type	Values	What it means
<b>ChangeMode</b>	BITMAP CMYK GRAYSCALE INDEXEDCOLOR LAB MULTICHANNEL RGB	The type of color mode to use.  <b>Note:</b> Color images must be changed to GRAYSCALE mode before you can change them to BITMAP mode.
<b>ChannelType</b>	COMPONENT MASKEDAREA SELECTEDAREA SPOTCOLOR	The type of channel: COMPONENT: related to document color mode MASKEDAREA: Alpha channel where color indicates masked area SELECTEDAREA: Alpha channel where color indicates selected are SPOTCOLOR: Alpha channel to store a spot color.
<b>ColorBlendMode</b>	BEHIND CLEAR COLOR COLORBURN COLORDODGE DARKEN DIFFERENCE DISSOLVE EXCLUSION HARDLIGHT HARDMIXBLEND HUE LIGHTEN LINEARBURN LINEARDODGE LINEARLIGHT LUMINOSITY MULTIPLY NORMAL OVERLAY PINLIGHT SATURATION SCREEN SOFTLIGHT VIVIDLIGHT	Color blend mode type.
<b>ColorModel</b>	CMYK GRAYSCALE HSB LAB NONE RGB	The color model to use.
<b>ColorPicker</b>	ADOBE APPLE PLUGIN WINDOWS	The color picker to use.
<b>ColorProfile</b>	CUSTOM NONE WORKING	The color profile type to use to manage this document.

Constant type	Values	What it means
<b>ColorReductionType</b>	ADAPTIVE BLACKWHITE CUSTOM GRAYSCALE MACINTOSH PERCEPTUAL RESTRICTIVE SELECTIVE WINDOWS	The color reduction algorithm option to use.
<b>ColorSpaceType</b>	ADOBERGB COLORMATCHRGB PROPHOTORGB SRGB	The type of color space to use.
<b>CopyrightedType</b>	COPYRIGHTEDWORK PUBLICDOMAIN UNMARKED	The copyright status of the document.
<b>CreateFields</b>	DUPLICATION INTERPOLATION	The method to use for creating fields.
<b>CropToType</b>	ARTBOX BLEEDBOX BOUNDINGBOX CROPBOX MEDIABOX TRIMBOX	The style to use when cropping a page.
<b>DCSType</b>	COLORCOMPOSITE GRAYSCALECOMPOSITE NOCOMPOSITE	The DCS format to use:  COLORCOMPOSITE: Creates a color composite file in addition to DCS files; GRAYSCALECOMPOSITE: Creates a grayscale composite file in addition to DCS files; NOCOMPOSITE: Does not create a composite file.
<b>DepthMapSource</b>	IMAGEHIGHLIGHT LAYERMASK NONE TRANSPARENCYCHANNEL	The source to use for the depth map.
<b>DescValueType</b>	ALIASTYPE BOOLEANVALUE CLASSTYPE DOUBLETYPE ENUMERATEDTYPE INTEGERTYPE LISTTYPE OBJECTTYPE RAWTYPE REFERENCETYPE STRINGTYPE UNITDOUBLE	The value type of an object.
<b>DialogModes</b>	ALL ERROR NO	Controls the type (mode) of dialogs Photoshop displays when running scripts.
<b>Direction</b>	HORIZONTAL VERTICAL	The orientation of the object.

Constant type	Values	What it means
<b>DisplacementMapType</b>	STRETCHTOFIT TILE	Describes how the displacement map fits the image if the image is not the same size as the map.
<b>Dither</b>	DIFFUSION NOISE NONE PATTERN	The type of dithering to use.
<b>DocumentFill</b>	BACKGROUNDCOLOR TRANSPARENT WHITE	The fill of the document.
<b>DocumentMode</b>	BITMAP CMYK DUOTONE GRAYSCALE INDEXEDCOLOR LAB MULTICHANNEL RGB	The color mode of the open document.
<b>EditLogItemsType</b>	CONCISE DETAILED SESSIONONLY	The history log edit options: CONCISE: Save a concise history log. DETAILED: Save a detailed history log. SESSIONONLY: Save history log only for the session.
<b>ElementPlacement</b>	INSIDE PLACEATBEGINNING PLACEATEND PLACEBEFORE PLACEAFTER	The object's position in the Layers palette.  <b>Note:</b> Not all values are valid for all object types. Please refer to the object property definition in <a href="#">JavaScript Object Reference</a> to make sure you are using a valid value.
<b>EliminateFields</b>	EVENFIELDS ODDFIELDS	The type of fields to eliminate.
<b>ExportType</b>	ILLUSTRATORPATHS SAVEFORWEB	The export options to use.
<b>Extension</b>	LOWERCASE NONE UPPERCASE	The formatting of the extension in the filename.



Constant type	Values	What it means
<b>FileNamingType</b>	DDMM DDMMYY DOCUMENTNAMELOWER DOCUMENTNAMEMIXED DOCUMENTNAMEUPPER EXTENSIONLOWER EXTENSIONUPPER MMDD MMDDYY SERIALLETTERLOWER SERIALLETTERUPPER SERIALNUMBER1 SERIALNUMBER2 SERIALNUMBER3 SERIALNUMBER4 YYDDMM YYMMDD YYYYMMDD	File naming options for the batch command.
<b>FontPreviewType</b>	LARGE MEDIUM NONE SMALL	The type size to use for font previews in the type tool font menus.
<b>ForcedColors</b>	BLACKWHITE NONE PRIMARIES WEB	The type of colors to be forced (included) into the color table: BLACKWHITE: Pure black and pure white; NONE; PRIMARIES: Red, green, blue, cyan, magenta, yellow, black, and white; WEB: the 216 web-safe colors.
<b>FormatOptions</b>	OPTIMIZEDBASELINE PROGRESSIVE STANDARDBASELINE	The option with which to save a JPEG file: OPTIMIZEDBASELINE: Optimized color and a slightly reduced file size; PROGRESSIVE: Displays a series of increasingly detailed scans as the image downloads; STANDARDBASELINE: Format recognized by most web browsers.
<b>GalleryConstrainType</b>	CONSTRAINBOTH CONSTRAINHEIGHT CONSTRAINWIDTH	The type of proportions to constrain for images.
<b>GalleryFontType</b>	ARIAL COURIERNEW HELVETICA TIMESNEWROMAN	The fonts to use for the Web photo gallery captions and other text.
<b>GallerySecurityTextColorType</b>	BLACK CUSTOM WHITE	The color to use for text displayed over gallery images as an antitheft deterrent.
<b>GallerySecurityTextPositionType</b>	CENTERED LOWERLEFT LOWERRIGHT UPPERLEFT UPPERRIGHT	The position of the text displayed over gallery images as an antitheft deterrent.

Constant type	Values	What it means
<b>GallerySecurityTextRotateType</b>	CLOCKWISE45 CLOCKWISE90 COUNTERCLOCKWISE45 COUNTERCLOCKWISE90 ZERO	The orientation of the text displayed over gallery images as an antitheft deterrent.
<b>GallerySecurityType</b>	CAPTION COPYRIGHT CREDIT CUSTOMTEXT FILENAME NONE TITLE	The content to use for text displayed over gallery images as an antitheft deterrent.  <b>Note:</b> All types draw from the image's file information except CUSTOMTEXT.
<b>GalleryThumbSizeType</b>	CUSTOM LARGE MEDIUM SMALL	The size of thumbnail images in the web photo gallery.
<b>Geometry</b>	HEPTAGON HEXAGON OCTAGON PENTAGON SQUARE TRIANGLE	Geometric options for shapes, such as the iris shape in the Lens Blur Filter.
<b>GridLineStyle</b>	DASHED DOTTED SOLID	The line style for the nonprinting grid displayed over images.
<b>GridSize</b>	LARGE MEDIUM NONE SMALL	The value of grid line spacing.
<b>GuideLineStyle</b>	DASHED SOLID	The line style for nonprinting guides displayed over images.
<b>IllustratorPathType</b>	ALLPATHS DOCUMENTBOUNDS NAMEDPATH	The paths to export.
<b>Intent</b>	ABSOLUTECOLORIMETRIC PERCEPTUAL RELATIVECOLORIMETRIC SATURATION	The rendering intent to use when converting from one color space to another.
<b>JavaScriptExecutionMode</b>	BEFORERUNNING NEVER ONRUNTIMEERROR	The debugger mode to use.
<b>Justification</b>	CENTER CENTERJUSTIFIED FULLYJUSTIFIED LEFT LEFTJUSTIFIED RIGHT RIGHTJUSTIFIED	The placement of paragraph text within the bounding box.

Constant type	Values	What it means
<b>Language</b>	BRAZILLIANPORTUGUES E CANADIANFRENCH DANISH DUTCH ENGLISHUK ENGLISHUSA FINNISH FRENCH GERMAN ITALIAN NORWEGIAN NYNORSKNORWEGIAN OLDGERMAN PORTUGUESE SPANISH SWEDISH SWISSGERMAN	The language to use.
<b>LayerCompression</b>	RLE ZIP	Compression methods for data for pixels in layers.
<b>LayerKind</b>	BRIGHTNESSCONTRAST CHANNELMIXER COLORBALANCE CURVES EXPOSURE GRADIENTFILL GRADIENTMAP HUESATURATION INVERSION LEVELS NORMAL PATTERNFILL PHOTOFILTER POSTERIZE SELECTIVECOLOR SMARTOBJECT SOLIDFILL TEXT THRESHOLD LAYER3D VIDEO	The kind of <code>artLayer</code> object.  <b>Note:</b> You can create a text layer only from an empty art layer.
<b>LensType</b>	MOVIEPRIME PRIME105 PRIME35 ZOOMLENS	The type of lens to use.
<b>MagnificationType</b>	ACTUALSIZE FITPAGE	The type of magnification to use when viewing an image.
<b>MatteType</b>	BACKGROUND BLACK FOREGROUND NETSCAPE NONE SEMIGRAY WHITE	The color to use for matting.
<b>MeasurementRange</b>	ALLMEASUREMENTS ACTIVEMEASUREMENTS	The measurement to take action upon

Constant type	Values	What it means
<b>MeasurementSource</b>	MEASURESELECTION MEASURECOUNTOOL MEASURERULERTOOL	The source for recording measurements
<b>NewDocumentMode</b>	BITMAP CMYK GRayscale LAB RGB	The color profile to use for the document.
<b>NoiseDistribution</b>	GAUSSIAN UNIFORM	Distribution method to use when applying an Add Noise filter.
<b>OffsetUndefinedAreas</b>	REPEATEDGEPIXELS SETTOBACKGROUND WRAPAROUND	Method to use to fill the empty space left by offsetting a an image or selection.
<b>OpenDocumentMode</b>	CMYK GRayscale LAB RGB	The color profile to use.
<b>OpenDocumentType</b>	ALIASPIX BMP CAMERARAW COMPUSEVEGIF DICOM ELECTRICIMAGE EPS EPSPICTPREVIEW EPSTIFFPREVIEW FILMSTRIP JPEG PCX PDF PHOTOCD PHOTOSHOP PHOTOSHOPDCS_1 PHOTOSHOPDCS_2 PHOTOSHOPEPS PHOTOSHOPPDF PICTFILEFORMAT PICTRESOURCEFORMAT PIXAR PNG PORTABLEBITMAP RAW SCITEXCT SGIRGB SOFTIMAGE TARGA TIFF WAVEFRONTSLA WIRELESSBITMAP	<p>The format in which to open the document.</p> <p><b>Note:</b> PHOTOCD is deprecated. Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS3 Install DVD.</p> <p><b>Note:</b> The DICOM option is for the Extended version only.</p>
<b>OperatingSystem</b>	OS2 WINDOWS	The operating system.
<b>Orientation</b>	LANDSCAPE PORTRAIT	The page orientation.

Constant type	Values	What it means
<b>OtherPaintingCursors</b>	PRECISEOTHER STANDARDOTHER	The pointer for the following tools: Eraser, Pencil, Paintbrush, Healing Brush, Rubber Stamp, Pattern Stamp, Smudge, Blur, Sharpen, Dodge, Burn, Sponge.
<b>PaintingCursors</b>	BRUSHSIZE PRECISE STANDARD	The pointer for the following tools: Marquee, Lasso, Polygonal Lasso, Magic Wand, Crop, Slice, Patch Eyedropper, Pen, Gradient, Line, Paint Bucket, Magnetic Lasso, Magnetic Pen, Freeform Pen, Measure, Color Sampler.
<b>Palette</b>	EXACT LOCALADAPTIVE LOCALPERCEPTUAL LOCALSELECTIVE MACOSPALETTE MASTERADAPTIVE MASTERPERCEPTUAL MASTERSELECTIVE PREVIOUSPALETTE UNIFORM WEBPALETTE WINDOWSPALETTE	The palette type to use.
<b>PathKind</b>	CLIPPINGPATH NORMALPATH TEXTMASK VECTORMASK WORKPATH	The type of path.
<b>PDFCompatibility</b>	PDF13 PDF14 PDF15 PDF16	The PDF version to make the document compatible with.
<b>PDFEncoding</b>	JPEG JPEG2000HIGH JPEG2000LOSSLESS JPEG2000LOW JPEG2000MED JPEG2000MEDHIGH JPEG2000MEDLOW JPEGHIGH JPEGLow JPEGMED JPEGMEDHIGH JPEGMEDLOW NONE PDFZIP PDFZIP4BIT	The type of compression to use when saving a document in PDF format.
<b>PDFResample</b>	NONE PDFAVERAGE PDFBICUBIC PDFSUBSAMPLE	The down sample method to use.

Constant type	Values	What it means
<b>PDFStandard</b>	NONE PDFX1A2001 PDFX1A2003 PDFX32002 PDFX32003	The PDF standard to make the document compatible with.
<b>PhotoCDColorSpace</b>	LAB16 LAB8 RGB16 RGB8	The color space to use when creating a Photo CD.  <b>Note:</b> Deprecated for Adobe Photoshop CS3. Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS3 Install DVD.
<b>PhotoCDSize</b>	EXTRALARGE LARGE MAXIMUM MEDIUM MINIMUM SMALL	The pixel dimensions of the image.  EXTRALARGE = 1024x1536 LARGE = 512x768 MAXIMUM = 2048x3072 MEDIUM = 256x384 MINIMUM = 64x96 SMALL = 128x192  <b>Note:</b> Deprecated for Adobe Photoshop CS3. Kodak PhotoCD is now found in the Goodies folder on the Adobe Photoshop CS3 Install DVD.
<b>PICTBitsPerPixels</b>	EIGHT FOUR SIXTEEN THIRTYTWO TWO	The number of bits per pixel to use when compression a PICT file.  <b>Note:</b> Use 16 or 32 for RGB images; use 2, 4, or 8 for bitmap and grayscale images.
<b>PICTCompression</b>	JPEGHIGHPICT JPEGLOWPICT JPEGMAXIMUMPICT JPEGMEDIUMPICT NONE	The type of compression to use when saving an image as a PICT file.
<b>PicturePackageTextType</b>	CAPTION COPYRIGHT CREDIT FILENAME NONE ORIGIN USER	The function or meaning of text in a Picture Package.
<b>PointKind</b>	CORNERPOINT SMOOTHPOINT	The role a <code>pathPoint</code> plays in a <code>pathItem</code> .
<b>PointType</b>	POSTSCRIPT TRADITIONAL	The kind of measurement to use for type points: POSTSCRIPT = 72 points/inch; TRADITIONAL = 72.27 points/inch.

Constant type	Values	What it means
<b>PolarConversionType</b>	POLARTORECTANGULAR RECTANGULARTOPOLAR	The method of polar distortion to use.
<b>Preview</b>	EIGHTBITTIFF MACOSEIGHTBIT MACOSJPEG MACOSMONOCHROME MONOCHROMETIFF NONE	The type of image to use as a low-resolution preview in the destination application.
<b>PrintEncoding</b>	ASCII BINARY JPEG	The type of encoding to use.
<b>PurgeTarget</b>	ALLCACHES CLIPBOARDCACHE HISTORYCACHES UNDOCACHES	Cache to be targeted in a purge operation.
<b>QueryStateType</b>	ALWAYS ASK NEVER	Permission state for queries.
<b>RadialBlurMethod</b>	SPIN ZOOM	The blur method to use.
<b>RadialBlurQuality</b>	BEST DRAFT GOOD	The smoothness or graininess of the blurred image.
<b>RasterizeType</b>	ENTIRELAYER FILLCONTENT LAYERCLIPPINGPATH LINKEDLAYERS SHAPE TEXTCONTENTS	The layer element to rasterize.
<b>ReferenceFormType</b>	CLASSTYPE ENUMERATED IDENTIFIER INDEX NAME OFFSET PROPERTY	The type of an <a href="#">ActionReference</a> object.
<b>ResampleMethod</b>	BICUBIC BICUBICSHARPER BICUBICSMOOTHER BILINEAR NEARESTNEIGHBOR NONE	The method to use for image interpolation.
<b>ResetTarget</b>	ALLTOOLS ALLWARNINGS EVERYTHING	The type of object or objects to reset to default settings.
<b>RippleSize</b>	LARGE MEDIUM SMALL	The size of undulations to use.
<b>SaveBehavior</b>	ALWAYS SAVE ASKWHENSAVING NEVERSAVE	The application's behavior when a <code>save()</code> method is called.

Constant type	Values	What it means
<b>SaveDocumentType</b>	ALIASPIX BMP COMPUSERVEGIF ELECTRICIMAGE JPEG PCX PHOTOSHOP PHOTOSHOPDCS_1 PHOTOSHOPDCS_2 PHOTOSHOPEPS PHOTOSHOPPDF PICTFileFORMAT PICTRESOURCEFORMAT PIXAR PNG PORTABLEBITMAP RAW SCITEXT SGIRGB SOFTIMAGE TARGA TIFF WAVEFRONT WIRELESSBITMAP	<p>The format in which to save a document.</p> <p><b>Note:</b> The <code>format</code> property of the <code>ExportOptionsSaveForWeb</code> class uses only the following values: <code>COMPUSERVEGIF</code>, <code>JPEG</code>, <code>PNG-8</code>, <code>PNG-24</code>, and <code>BMP</code>. See <a href="#">ExportOptionsSaveForWeb</a>.</p>
<b>SaveEncoding</b>	ASCII BINARY JPEGHIGH JPEGLOW JPEGMAXIMUM JPEGMEDIUM	The type of encoding to use when saving a file.
<b>SaveLogItemsType</b>	LOGFILE LOGFILEANDMETADATA METADATA	The location of history log data.
<b>SaveOptions</b>	DONOTSAVECHANGES PROMPTTOSAVECHANGES SAVECHANGES	The 'save' method to use when closing a document.
<b>SelectionType</b>	DIMINISH EXTEND INTERSECT REPLACE	<p>The selection behavior when a selection already exists: <code>DIMINISH</code>: Remove the selection from the already selected area; <code>EXTEND</code>: Add the selection to an already selected area; <code>INTERSECT</code>: Make the selection only the area where the new selection intersects the already selected area; <code>REPLACE</code>: Replace the selected area.</p>
<b>ShapeOperation</b>	SHAPEADD SHAPEINTERSECT SHAPESUBTRACT SHAPEXOR	A <code>subPathItem</code> object's behavior when it intersects another <code>subPathItem</code> object.
<b>SmartBlurMode</b>	EDGEONLY NORMAL OVERLAYEDGE	<p>The method to use for smart blurring: <code>EDGEONLY</code>, <code>OVERLAYEDGES</code>: Apply blur only to edges of color transitions; <code>NORMAL</code>: Apply blur to entire image.</p>



Constant type	Values	What it means
<b>SmartBlurQuality</b>	HIGH LOW MEDIUM	The blur quality to use.
<b>SourceSpaceType</b>	DOCUMENT PROOF	The color space for source when printing.
<b>SpherizeMode</b>	HORIZONTAL NORMAL VERTICAL	The curve (or stretch shape) to use for the distortion.
<b>StrikeThruType</b>	STRIKEBOX STRIKEHEIGHT STRIKEOFF	The style of strikethrough to use.
<b>StrokeLocation</b>	CENTER INSIDE OUTSIDE	The placement of path or selection boundary strokes.
<b>TargaBitsPerPixels</b>	SIXTEEN THIRTYTWO TWENTYFOUR	The resolution to use when saving an image in Targa format.
<b>TextCase</b>	ALLCAPS NORMAL SMALLCAPS	The case usage for type.
<b>TextComposer</b>	ADOBEEVERYLINE ADOBESINGLELINE	The composition method to use to optimize the specified hyphenation and justification options.
<b>TextType</b>	PARAGRAPHTEXT POINTTEXT	The type of text: PARAGRAPHTEXT: Text that wraps within a bounding box; POINTTEXT: Text that does not wrap.
<b>TextureType</b>	BLOCKS CANVAS FILE FROSTED TINYLENS	The type of texture or glass surface image to load for a texturizer or glass filter.
<b>TIFFEncoding</b>	JPEG NONE TIFFLZW TIFFZIP	The encoding to use for TIFF files.

Constant type	Values	What it means
<b>ToolType</b>	ARTHISTORYBRUSH BACKGROUNDERASER BLUR BRUSH BURN CLONESTAMP COLORREPLACEMENTTOOL DODGE ERASER HEALINGBRUSH HISTORYBRUSH PATTERNSTAMP PENCIL SHARPEN SMUDGE SPONGE	The tool selection.
<b>TransitionType</b>	BLINDSHORIZONTAL BLINDSVERTICAL BOXIN BOXOUT DISSOLVE GLITTERDOWN GLITTERRIGHT GLITTERRIGHTDOWN NONE RANDOM SPLITHORIZONTALIN SPLITHORIZONTALOUT SPLITVERTICALIN SPLITVERTICALOUT WIPEDOWN WIPELEFT WIPERIGHT WIPEUP	The method to use to transition from one image to the next in a PDF presentation.
<b>TrimType</b>	BOTTOMRIGHT TOPLEFT TRANSPARENT	Type of pixels to trim around an image: BOTTOMRIGHT = bottom right pixel color; TOPLEFT = top left pixel color.
<b>TypeUnits</b>	MM PIXELS POINTS	The unit to use for measuring text characters.
<b>UndefinedAreas</b>	REPEATEDGEPIXELS WRAPAROUND	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.
<b>UnderlineType</b>	UNDERLINELEFT UNDERLINEOFF UNDERLINERIGHT	The placement of text underlining.  <b>Note:</b> UnderlineType.UNDERLINELEFT and UnderlineType.UNDERLINERIGHT are valid only when direction = Direction.VERTICAL.

Constant type	Values	What it means
<b>Units</b>	CM INCHES MM PERCENT PICAS PIXELS POINTS	The measurement unit for type and ruler increments.
<b>Urgency</b>	FOUR HIGH LOW NONE NORMAL SEVEN SIX THREE TWO	The editorial urgency of the artwork.
<b>WarpStyle</b>	ARC ARCH ARCLOWER ARCUPPER BULGE FISH FISHEYE FLAG INFLATE NONE RISE SHELLLOWER SHELLUPPER SQUEEZE TWIST WAVE	The warp style to use.
<b>WaveType</b>	SINE SQUARE TRIANGULAR	The type of wave to use.
<b>WhiteBalanceType</b>	ASSHOT AUTO CLOUDY CUSTOM DAYLIGHT FLASH FLUORESCENT SHADE TUNGSTEN	The lighting conditions to use (affects color balance).
<b>ZigZagType</b>	AROUNDCENTER OUTFROMCENTER PONDRIPPLES	The method of zigzagging to use.

This section describes the JavaScript resource that enables your JavaScripts to behave like a plug-in. This includes:

- the ability to specify a menu the script appears in as a command,
- a terminology resource so the script can function with the Action Manager, which allows your script to record and be automated by scripting parameters,
- a category to enable ordering and grouping of commands within menus, and
- an enable string that indicates whether the command is enabled or disabled given a set of conditions.

## JavaScript Resource Syntax

The JavaScript Resource has an HTML-style syntax, with each `<tag>` matched by a closing `</tag>`. This resource needs to appear within comments ( `/* ... */` ) and should be defined at the top of your script file (within the first 10,240 characters of the file.)

Tag	Description
<code>&lt;javascriptresource&gt;</code>	The resource definition tag.
<code>&lt;name&gt;</code>	The command name that appears in the Photoshop menu. If this tag is not provided in the resource, the name of the command in the menu defaults to the name of the script.
<code>&lt;type&gt;</code>	The menu the command appears in. If this tag is not provided, the command appears in the <b>File &gt; Scripts</b> menu.  <b>Note:</b> Currently the only supported value for <code>&lt;type&gt;</code> is <code>automate</code> . This value puts the script in the <b>File &gt; Automate</b> menu.
<code>&lt;about&gt;</code>	A string that appears in an About box, which the user can select from the <b>Help &gt; About Plug-in</b> menu.
<code>&lt;enableinfo&gt;</code>	A boolean expression that indicates whether the command is enabled in the menu. See <a href="#">Enable Info Grammar</a> .  <b>Note:</b> If you provide this tag, the menu item is enabled if and only if there is at least one document open, and the boolean expression evaluates to true. If you always want the menu item enabled, do not use this tag.
<code>&lt;eventid&gt;</code>	A unique string that identifies the event. Using a UUID will ensure that your script won't share this identifier with another script.

Tag	Description
<category>	The category the command appears within in the menu. Used to group and order commands in the menu. Commands are placed in the menu alphabetically based on the string in <category>. If two commands use the same category, they are grouped together.
<terminology>	The terminology dictionary for the script to function with the Action Manager. See the <a href="#">Terminology Dictionary</a> .

## Basic JavaScript Resource Example

This example shows a very basic <javascriptresource>. With this resource, the script can be executed by selecting the command **Add a Document**, which appears in the **Automate** menu. This command is enabled in the menu, provided at least one document is already open. If the user requests information about the script from the **About Plug-in** menu, the string contained in the <about> tag is displayed in a dialog box.

```
/*
  <javascriptresource>
  <name>Add a Document</name>
  <type>automate</type>
  <about>A short string providing information about the script.</about>
  <enableinfo>true</enableinfo>
</javascriptresource>
*/

app.documents.add();
```

## Enable Info Grammar

The <enableinfo> tag provides a boolean expression that, when evaluated, indicates whether the command is enabled in the menu. You can use this expression to enable or disable the menu item based on various characteristics of the document. The Enable Info grammar is as follows:

```
<booleanExpression> := <conjunction> { "|" <conjunction> }
<conjunction> := <relation> {"&&" <relation> }
<relation> := <equality> {<relationOperator><equality>}
<equality> := <simpleExpression> {<equalityOperator><simpleExpression>}
<simpleExpression> := <term> {<addOperator><term>}
<term> := <factor> {<mulOperator><factor>}
<factor> := <integer> | <intrinsic> | <ident> |
  "(" <booleanExpression> ")" | "(" simpleExpression ")" |
  "+" <factor> | "-" <factor> | "!" <factor>
<integer> := digit {digit}
<intrinsic> := <limitFunction> | <dimFunction> | <inFunction>
<limitFunction> := ( "min" | "max" ) "(" <simpleExpression> ","
  <simpleExpression> { "," <simpleExpression> } ")"
<dimFunction> := "dim" "(" <simpleExpression> "," <simpleExpression> ")"
```

```

<ident> :=          (alpha | "_" ) {alpha | digit | "_" }
<mulOperator> :=    "*" | "/"
<addOperator> :=    "+" | "-"
<equalityOperator> :=  "==" | "!="
<relationOperator> := "<" | "<=" | ">=" | ">"
<inFunction> :=      "in" "(" <simpleExpression> {"," <simpleExpression> } ")"

```

Operator precedence is shown in the following table. Operators are listed with the highest order of precedence at the top of the table

Operator	Description
	Or
&&	And
+ -	Addition or subtraction
* /	Multiply or divide
< <= >= >	Less than, less than or equal, greater than or equal, greater than
== !=	Equals, or does not equal.
(..) in() max() min() unary + - !	Functions Unary operators: increment, decrement, not

The grammar provides variables and constants that you can use in the `<enableinfo>` expression. The following table provides a list of the constants that are available.

Constant Name	Description
true	Boolean true
false	Boolean false
BitmapMode	Bitmap mode.
GrayscaleMode	Grayscale mode, 8 bit depth.
IndexedMode	Indexed color mode.
RGBMode	RGB color mode.
CMYKMode	CMYK color mode.
HSLMode	HSL color mode.
HSBMode	HSB color mode
MultiChannelMode	Multichannel mode.
DuotoneMode	Duotone mode.
LabMode	Lab color mode.

Constant Name	Description
Gray16Mode	Grayscale mode, 16 bits per channel
RGB48Mode	RGB color mode, 16 bits per channel.
Lab48Mode	LAB mode, 16 bits per channel.
CMYK64Mode	CMYK mode, 16 bits per channel.
DeepMultichannelMode	Deep multichannel mode.
Duotone16Mode	Duotone mode, 16 bit depth.
RGB96Mode	RGB color mode, 32 bits per channel.
Gray32Mode	Grayscale mode, 32 bit depth.

The following table show the set of variables you can use in the <enableinfo> expression. The value of these variables is set based on the properties of the active document.

Variable Name	Description
PSHOP_ImageMode	Image mode of the active document.
PSHOP_ImageDepth	Depth of the active document.
PSHOP_HasLayerMask	Boolean indicating presence of layer mask.
PSHOP_HasSelectionMask	Boolean indicating presence of selection mask.
PSHOP_HasTransparencyMask	Boolean indicating presence of transparency mask.
PSHOP_NumTargetChannels	Number of target channels.
PSHOP_NumTrueChannels	Numer of image channels.
PSHOP_IsAdjustorSheet	Boolean
PSHOP_IsTargetComposite	Boolean indicating whether channels are flattened.
PSHOP_IsTargetSection	Boolean.
PSHOP_IsTargetVisible	Boolean.
PSHOP_ImageWidth	Width of the image.
PSHOP_ImageHeight	Height of the image.
PSHOP_TargetProtectFlags.	

## Undefined Values in Enable Info Evaluation

If any arithmetic or relation operation contains an operand whose value is undefined, or a variable that is undefined, the result of that evaluation is `false`.

Boolean values are treated as in C/C++, where non-zero values are `true`, and zero is `false`, with the exception that an undefined value is also `false`.

## Using the “in” Function

The `in` function (see `<inFunction>`) returns `true` if the first parameter is equal to at least one of the subsequent parameters. A typical use might be to see if the image mode of the active document is one of a set of image modes. For example:

```
in(PSHOP_ImageMode, RGBMode, CMYKMode, LabMode)
```

## Action Manager Automation

For your script to be able to record scripting parameters and be automated by them, it requires the addition of two basic mechanisms:

- A *terminology dictionary* that maps your script’s user interface to human readable text, providing text and type information for each parameter the script uses.
- Code to read parameter information when it comes from the Action Manager, rather than from the user-interface, and code to write parameter information to the Action Manager. This code uses the Action Manager classes [ActionDescriptor](#), [ActionList](#), and [ActionReference](#).

See `Conditional Mode Change.jsx` for an example of a script that can record and be automated by scripting parameters.

## Terminology Dictionary

The JavaScript resource provides a `<terminology>` tag that allows you to provide the terminology dictionary for your script. The first step in creating a terminology dictionary is to review your script’s user interface, and create human-readable strings for each element in your user interface.

For example, in the Conditional Mode Change command, the user interface requests a source mode and a target mode. Both source mode and target mode have several options. All of these elements of the user interface need to have entries in the terminology dictionary.

The terminology dictionary is created in a PDF dictionary format, with the following entries, and must have the following format in the `<javascriptresource>`:

```
<terminology><![CDATA[<<<
  /Version integer
  /Events <<event dictionary>>
  /Classes <<class dictionary>>
  /Enumerations <<enumeration dictionary>>
>>> ]]></terminology>
```

**Note:** The information in the terminology tag needs to be wrapped in a `CDATA` block so the xml parser will ignore “/” and other tags that appear in the terminology.

The definitions for events, classes and enumerations dictionaries are provided below.



The `/Events` dictionary contains an entry for each event:

```

/eventName [                                // Name used in string-based API
  (String event name)                       // required
  /direct parameter type                   // optional; if omitted, no parameter
  <<                                       // optional parameter dictionary
    /parameterName [                       // Name used in string-based API
      (String name)                       // required
      /parameter type                     // required
    ]
    ...                                    // other parameters
  >>
]

```

The `/Classes` dictionary contains an entry for each class:

```

/className [                                // Name used in string-based API
  (ZString class name)                   // required
  <<                                       // property dictionary
    /propertyName [                       // Name used in string-based API
      (String name)                       // required
      /property type                     // required
    ]
    ...                                    // other properties
  >>
]

```

The `/Enumerations` dictionary contains an entry for each enumerated type:

```

/enumTypeName                               // Name used in string-based API
<<
  /enumValue (String name)               // required
  ...
>>

```

## Value Type Definitions

For `/parameter type` and `/property type` definitions, you can use the Class and Enumeration type declarations you make in your own terminology dictionary, you can use declarations provided by Photoshop or you can use basic value types.

### Basic Value Types

The basic value types are shown in the following table:

Name	Code	Description
<code>typeInteger</code>	<code>'long'</code>	<code>int32</code>
<code>typeFloat</code>	<code>'doub'</code>	IEEE 64 bit double
<code>typeBoolean</code>	<code>'bool'</code>	TRUE OR FALSE.
<code>typeText</code>	<code>'TEXT'</code>	Block of any number of readable characters.
<code>typeAlias</code>	<code>'alis'</code>	Macintosh file system path.

Name	Code	Description
typePaths	'Pth '	Windows file system path.
typePlatformFilePath	'alis' or 'Pth	typeAlias for Mac OS, typePath for Windows.

### Predefined Class Types

Photoshop provides a number of predefined classes that are available for use in the terminology dictionary. A useful subset of those classes is shown in the table below. Use these classes when they are appropriate, but you can define new classes in the terminology resource, if necessary.

Name	Code	Description
classColor	'Clr '	Class for color classes.
classRGBColor	'RGBC'	keyRed, keyGreen, keyBlue
classCMYKColor	'CMYC'	keyCyan, keyMagenta, keyYellow, keyBlack.
classUnspecifiedColor	'UnsC'	Unspecified.
classGrayscale	'Grsc'	keyGray
classBookColor	'BkCl '	Book color
classLabColor	'LbCl '	keyLuminance, keyA, keyB.

## Uniqueness Rules for Terminology Entries

Generally, the names for terminology entries must be unique within a particular category and scope. It is best to not make names unique unnecessarily; generic terms are preferable, and if a name already exists for something, go ahead and use it. Case matters in considering uniqueness of terminology entries.

The uniqueness rules for terminology entries are:

- All event names must be different from all other event names.
- All class names must be different from all other names.
- All enumeration type names must be different from all other enumeration type names.
- All keys must be different from all other keys used in the same class or event.
- All enumeration values must be different from all other enumeration values in the same enumeration type.
- A class, event, enumeration type, key, and enumeration value can all have the same name.

## Terminology Definition Example

This example demonstrates the terminology definition for a new event; the example uses ZStrings. The event is called `newAnnot`, and it takes three parameters:

- `annotType`, an enumeration (`annotType`)
- `at`, a class (`point`), and
- `size`, a class (`annotSizeClass`).

The `annotSizeClass` has two properties: `width`, and `height`, both of type `floatType`. The enumeration `annotType` has three values: `annotUnknown`, `annotText`, and `annotSound`.

```
<terminology><![CDATA[<<<
  /Version 1
  /Events
    /newAnnot [(New Annotation) <<
      /annotType [(Type) /annotType]
      /at [(At) /Point]
      /size [(Size) /annotSizeClass] >>]
  /Classes
    /annotSizeClass [(Size) <<
      /width [(Width) /floatType]
      /height [(Height) /floatType]
      >>]
  /Enumerations
    /annotType <<
      /annotUnknown (Unknown)
      /annotText (Text)
      /annotSound (Sound)
    >>> ]]></terminology>
```

## 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.

**Tip:** If you can't find the event you want to use for notification in this table, you can use `ScriptListener` to determine the event ID code. See the `ScriptListener` documentation in the Action Manager chapter of the *Photoshop CS3 Scripting Guide*.

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'

Event	4-char ID or String
Charcoal	'Chrc'
Chrome	'Chrm'
Clear	'Cler'
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'

Event	4-char ID or String
Deselect	'Dslc'
Despeckle	'Dspc'
DifferenceClouds	'DrfC'
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'

Event	4-char ID or String
GaussianBlur	'GsnB'
Get	'getd'
Glass	'Gls '
GlowingEdges	'GlwE'
Gradient	'Grdn'
GradientMap	'GrMp'
Grain	'Grn '
GraphicPen	'GraP'
Group	'GrpL'
Grow	'Grow'
HalftoneScreen	'Hlfs'
Hide	'Hd '
HighPass	'HghP'
HSBHSL	'HsbP'
HueSaturation	'HStr'
ImageSize	'ImgS'
Import	'Impr'
InkOutlines	'InkO'
Intersect	'Intr'
IntersectWith	'IntW'
Inverse	'Invs'
Invert	'Invr'
LensFlare	'LnsF'
Levels	'Lvls'
LightingEffects	'LghE'
Link	'Lnk '
Make	'Mk '
Maximum	'Mxm '
Median	'Mdn '
MergeLayers	'Mrg2'
MergeLayersOld	'MrgL'
MergeSpotChannel	'MSpt'
MergeVisible	'MrgV'

Event	4-char ID or String
Mezzotint	'Mztn'
Minimum	'Mnm '
ModeChange	"8cba8cd6-cb66-11d1-bc43-0060b0a13dc4"
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 '



Event	4-char ID or String
PosterEdges	'PstE'
Posterize	'Pstr'
Previous	'Prvs'
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'

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

Event	4-char ID or String
WaterPaper	'WtrP'
Watercolor	'Wtrc'
Wave	'Wave'
Wind	'Wnd '
ZigZag	'ZgZg'
BackLight	'BacL'
FillFlash	'File'
ColorCast	'ColE'

# Index

## A

- Action Manager 208
- actions
  - command lists 40
  - descriptions 43
  - descriptors 37
  - playing 47
- active document 45
- Add Noise filter
- adjustments
  - brightness 54
  - color 188
  - color balance 54, 62
  - contrast 54, 60
  - curves 55
  - highlights 62
  - levels 55, 60
  - shadows 62
  - temperature 62
- Adobe Illustrator, exporting paths to 105
- alpha channels
  - defined 72
  - from transparency (TIFF documents) 186
  - opacity 72
  - saving
    - in BMP documents 69
    - in PDF documents 148
    - in PICT documents 153
    - in PICT resources 154
    - in Pixar documents 156
    - in PSD documents 152
    - in RAW documents 164
    - in SGIRGB documents 171
    - in Targa documents 176
    - in TIFF documents 186
- anchor points
  - adding 144
  - specifying position of 188
- annotations, importing 93
- anti aliasing
  - options 188
  - text 179
- application
  - activating 47
  - checking if feature enabled 48
  - defaults 158
  - location 46
  - preferences 158
- artLayers, *See* layers
- Asian text 161
- authors 98
- auto kerning 179, 188
- auto leading 184
- auto spacing, contact sheets 82
- available memory 45
- Average filter 55

## B

- background color
  - application 45
  - galleries 109
- background layers 53
- baseline shift 179
- batch command 47
- batches
  - destination folder 66, 188
  - specifying options 66
- beeping 158
- bitmap documents
  - converting to 188
  - depth type 189
  - halftone type 188
  - opening 196
  - saving 69
- bitmap images
  - See* bitmap documents
- black and white images 63
- blending modes
  - layer sets 128
  - layers 53
  - options 189
- Blur filter 55
- blur filters
  - Average 55
  - Blur More 55
  - Gaussian Blur 56
  - Lens Blur 57
  - Motion Blur 58
  - Radial Blur 58
  - Smart Blur 59
- Blur More filter 55
- BMP documents
  - See* bitmap documents
- brightness 54
  - adjusting 54
  - equalizing 60

## C

- caches
  - images 159
  - purging 49
- camera raw documents
  - opening 70
  - settings 189
  - size options 189
- canvas
  - flipping 93
  - resizing 94
- canvas, defined 89
- captions
  - contact sheets 82
  - documents 98
  - gallery images 110
  - gallery thumbnails 114
  - images 98
- channels

- activating 89
- adding 74
- adjusting 55
- alpha *See* alpha channels
- creating 74
- deleting 73
- displaying in color 158
- duplicating 73
- making visible 73
- merging 73
- mixing 61
- splitting 95
- spot *See* spot channels
- types of 72
- clipping paths
  - from paths 140
  - from text 185
- Clouds filter 55
- CMYKColor 79
- color balance, adjusting 62
- color picker 158
- color profiles
  - changing 92
  - determining type of 90
  - naming 89
- color profiles, *see* individual document formats
- color samplers
  - adding 81
  - creating 81
  - moving 80
  - removing 80
- colors
  - active links 109
  - adjusting 188
  - balancing 54
  - channels 72
  - CMYK 79
  - custom settings 112
  - in galleries 109
  - inverting 60
  - modifying 62
  - none 134
  - preserving (GIF only) 115
  - reduction 106
  - settings 45
  - solid color objects 172
  - testing if equal 172
  - visited links 109
- comments, layer comps 124
- compatibility, maximizing 160
- component channels
  - color balance 54
  - defined 72
  - listing 89
  - See* composite channels
- composite channels 72
  - See* component channels
- Compuserve GIF documents
  - opening 196
  - saving 115

- contact sheets
  - captions 82
  - columns 82
  - dimensions 83
  - making 48
  - rows 82
- contrast
  - adjusting 54
  - adjusting automatically 60
  - camera raw settings 70
  - midtones 62
- copyrights 98
- count items
  - adding 85
  - creating 85
  - removing 84
- cropping 92
- CS3 version changes 32
- cursors 160
- curves, adjusting 55
- Custom filters 55

## D

- DCS 1 documents, saving 86
- De-Interlace filter 55
- desaturate 60
- Despeckle filter 55
- dialogs
  - displaying 45
- Difference Clouds filter 55
- Diffuse Glow filter 55
- Displace filter 56
- distort filters
  - Diffuse Glow 55
  - Displace 56
  - Glass Effect 56
  - Ocean Ripple 58
  - Pinch 58
  - Polar Coordinates 58
  - Ripple 58
  - Shear 59
  - Spherize 59
  - Twirl 59
  - Wave 59
  - Zigzag 59, 60
- document formats, *see* individual document formats
- DocumentInfo 98
- documents 89
  - activating 45
  - adding 102
  - closing 91
  - code sample 96
  - color profiles 89
  - color samplers 90
  - counting items 90
  - counting objects 92
  - cropping 93
  - dimensions 89
  - duplicating 93

- exporting 93
- info 98
- loading 48
- managed 90
- measurement scale 90
- metadata 90, 98
- open with Photoshop dialog 49
- opening 49
- optimizing for web 106
- printing 94
- resizing 94
- resolution 91
- saving 94, 95
- suspending history 95
- trapping (CMYK) 95
- trimming 96
- Dust and Scratches filter 56

**E**

- Enable Info
  - constants 206
  - grammar 205
  - operator precedence 206
  - variables 207
- EPSSaveOptions 104
- equalize 60
- event IDs
  - using ScriptListener to find 212
- Events Manager 46
- executing scripts 31, 32
- exif 98
- exporting
  - documents 93
  - paths 105
  - to Illustrator 105
  - to Web 106
- ExportOptionsIllustrator 105
- ExportOptionsSaveForWeb 106

**F**

- file extensions
  - format 161
  - including 158
  - script files 31
- file metadata 98
- files
  - merging 48
- filetypes
  - macOS 45
  - Windows 46
- filling
  - paths 139
  - selections 167
- filter, *see* individual filter names
- Folder object 31
- fonts
  - detecting 45
  - determining family of 177
  - determining style of 177

- PostScript name of 177
- formats, *see* individual document formats

**G**

- galleries 112
  - background color 109
  - banners 108
  - captions 110
  - color options 109
  - credits 110
  - dimensions 110
  - filenames 110
  - link colors 109
  - making 47, 48
  - metadata 112
  - photographer 108
  - security text 113
  - thumbnail images 114
- GalleryBannerOptions 108
- GalleryCustomColorOptions 109
- GalleryImagesOptions 110, 111
- GalleryOptions 112
- GallerySecurityOptions 113
- GalleryThumbnailOptions 114
- Gaussian Blur filter 56
- GIF documents
  - See* Compuserve GIF documents
- GIFSaveOptions 116
- Glass Effect filter 56
- glyph scaling 179–183
- GrayColor 117
- grids 159
- grouped layers 53
- guides 159

**H**

- halftone screen 68
- hanging punctuation 180
- High Pass filter 56
- highlights
  - adjusting 62
  - color balance 54
- histograms
  - channels 72
- history log 161
- history states
  - activating 89
  - allowing nonlinear 160
  - default number of 160
  - snapshot 118
  - suspending 95
- HSBColor 120
- hyphenation 181

**I**

- IDs
  - getting 37
  - PICT Resource 154

- property 43
- runtime 47
- runtime to string 50
- string to runtime 49
- string to type 49
- type to char 49
- Illustrator
  - See Adobe Illustrator
- image
  - resizing 94
- image pyramids 186
- images
  - bitmap 68
  - black and white 63
  - caches 159
  - captions 110
  - definition of 89
  - desaturating 60
  - equalizing 60
  - filetypes 45
  - from split channels 95
  - inverting colors 60
  - previewing 159
  - pyramids 186
  - resizing 94
  - resizing in galleries 111
  - thumbnails 114
- indexed color model 121
- IndexedConversionOptions 121
- individual document formats, examples 95
- installing scripts 32
- interpolation 159

**J**

- JavaScript
  - changes in Photoshop CS3 32
  - supported features 31
- JavaScript Resource
  - Enable Info grammar 205
  - javascriptresource syntax 204
- javascriptresource tag 204
- JPEG
  - quality 122
- JPEG documents
  - quality 122
  - saving 122
- JPEG options
  - scans 122
- JPEGSaveOptions 122
- justification 181

**K**

- kerning 179
  - text
    - auto kerning 188
- keyboard behavior 160

**L**

- LabColor 123
- languages 181
- layer comps 124
  - adding 126
  - applying 124
  - in documents 90
- layer sets
  - adding 130
  - art layers in 128
  - duplicating 129
  - in documents 90
  - linked layers in 128
  - linking 129
  - locking contents 128
  - moving 129
  - nesting 128
  - opacity 128
  - unlinking 129
- layer styles, applying 59
- LayerComps 126
- layered TIFFdocuments, saving 158
- Layers 127
- layers
  - adding 65
  - applying styles 59
  - background 53
  - blending mode 53
  - bounds 53
  - clipboard commands 60
  - comps 124
  - copying 60
  - duplicating 60
  - flattening 93
  - grouping 53
  - in documents 90
  - inverting 60
  - kind 53
  - linking 60
  - locking contents 53–54
  - making visible 54
  - merging 61
  - merging visible 93
  - moving 61
  - rasterizing 94
  - rasterizing contents 62
  - removing 65
  - resizing 62
  - rotating 62
  - saving in PDF documents 148
  - unlinking 63
- LayerSet 128
- LayerSets 130
- layersets
  - merging 129
- leading 181, 184
- Lens Blur filter
  - applying 57
- Lens Flare filter 57
- letter spacing 180–183

- levels
  - adjusting 55
  - adjusting automatically 54
- ligatures 179–181
- linked layers 60
  - unlinking 63
- links
  - colors 109

## M

- MacOS
  - filetypes 45
- managed documents 90
- maximizing compatibility 160
- Maximum filter 58
- Median Noise filter 58
- memory 45
- merging
  - layers 61
  - visible layers 93
- metadata
  - document 90
  - document object 98
  - galleries 112
  - xmp 91, 187
- methods
  - batch 46
- midtones
  - color balance 54
- Minimum filter 58
- Motion Blur filter 58

## N

- noise filters
  - Add Noise
  - Despeckle 55
  - Dust and Scratches 56
  - Median Noise 58
- nonlinear history 160
- notifications
  - events within scripts 135
- notifiers
  - adding 138
  - event IDs 212
  - removing 136
- NTSC filter 58

## O

- object model
  - changes in Photoshop CS3 32
- Ocean Ripple filter 58
- Offset filter 58
- old style type 183
- opacity
  - channels 72
  - gallery security text 113
  - layer fill 53
  - layer sets 128

- layers 54
- picture packages 155
- open options
  - Camera Raw format 70
  - DICOM format 88
  - EPS format 103
  - PDF format 147
  - Photo CD format 151
  - RAW format 163

- optimizing 106

- other filters

- Custom 55
  - High Pass 56
  - Maximum 58
  - Minimum 58
  - Offset 58

## P

- palettes 161
- pasting 94
- path items
  - adding 143
  - deselecting 139
  - filling 139
  - from text 185
  - making selection 140
  - path points 174
  - selecting 140
  - specifying path kind 139
  - stroking 140
  - sub items 139
  - sub path info 173
  - sub path items 174
  - work path from selection 167
- path point info
  - anchor points 145
  - left direction 145
  - right direction 145
- path points
  - anchor points 144
  - left direction 144
  - right direction 144
- PathItems 143
- paths
  - See *path items*
- PDF documents
  - opening 147
  - saving 148
- PDF presentations
  - auto advance 162
  - making 48
  - output format 162
  - transition type 162
- Photo CD discs, opening 151
- photo filtering 62
- photo galleries
  - See galleries
- photomerge 48
- Photoshop documents



- opening 196
- saving 152
- Photoshop files, maximizing compatibility 160
- PICT documents
  - opening 196
  - saving 153
- PICT resources
  - opening 196
  - saving 154
- picture packages
  - contents 155
  - flattening 155
  - making 48
  - opacity 155
  - options 155
  - text properties 155
- Pinch filter 58
- Pixar documents
  - opening 196
  - saving 156
- PixarSaveOptions 156
- pixels
  - aspect ratio 91
  - doubling 160
  - equalizing 60
  - interpolation 159
  - locking 54
  - unit measures 203
- playback options 46
- plug-in folder
  - additional plug-in folder 161
- PNG 8 documents, saving 107
- PNG documents
  - saving 133, 157
- PNGSaveOptions 157
- Polar Coordinates filter 58
- posterrize 62
- postscript encoding 94
- PostScript names 177
- Preferences 158
- PresentationOptions 162
- presentations
  - making 48
  - PDF presentations
- printing, documents 94
- property
  - measurementLog 45
- PSD documents
  - opening 196
  - saving 152
- purging 49

## Q

- quote style 161

## R

- Radial Blur filter 58
- rasterize 62
- rasterizing

- document layers 94
- RAW documents
  - opening 163
- RawSaveOptions 164
- render filters
  - Clouds 55
  - Difference Clouds 55
  - Lens Flare 57
- resolution
  - bitmap conversions 68
  - documents 91
- RGBColor 165
- Ripple filter 58
- rotation 62
- ruler units 160

## S

- save as 95
- saving 94
- saving, *see* individual document formats.
- scripting interface
  - build date 46
  - version 46
- scripts
  - automation 204, 208
  - enabling/disabling in menu 204
  - executing 31, 32
  - grouping in menu 204
  - installing 32
  - startup 32
  - terminology dictionary 208
  - valid file extensions 31
- Scripts Events Manager 46
- selected areas 91
- selections 166
  - boundaries 166
  - clearing 166
  - copying 166
  - cutting 166
  - deselecting 166
  - feathering 166
  - filling 167
  - from paths 140
  - making work path from 167
  - resizing 166, 167
  - rotating 167
  - smoothing 168
  - stroking 168
- selective color 62
- SGIRGB documents
  - saving 171, 200
- SGIRGBSaveOptions 171, 188
- shadows
  - adjusting 62
  - color balance 54
- Sharpen Edges filter 59
- Sharpen filter 58
- sharpen filters
  - Sharpen 58

- Sharpen Edges 59
- Sharpen More 59
- Unsharp Mask 59
- Sharpen More filter 59
- Shear filter 59
- Smart Blur filter 59
- smart quotes 161
- Spherize filter 59
- spot channels
  - defined 72
  - merging into component channels 73
  - opacity 72
  - saving
    - in DCS 2 documents 87
    - in PDF documents 149
    - in PSD documents 152
    - in RAW documents 164
    - in SGIRGB documents 171
    - in TIFF documents 186
- spotColors 171
- startup scripts 32
- strike thru 184
- stroking
  - default stroke color 45
  - path items 140
  - selections 168
- styles, applying 59
- sub path items 139

## T

- Targa documents, saving 176
- temperature 62
- terminology dictionary
  - defined 208
  - syntax 208
- text
  - Asian 161
  - auto kerning 179
  - auto leading 184
  - captions 110
  - color
  - composer 184
  - content 179
  - creating paths from 185
  - formatting 184
  - gallery security 113, 194
  - hyphenation 181
  - in picture packages 155
  - justification 181
  - languages 181
  - offset 179
  - orientation 180
  - spacing 180–183
  - tracking 184
  - wrapping 181
- text composer 184
- text fonts
  - See fonts

- text items
  - See text
- text layers
  - adding contents 179
  - creating 53
- Texture Fill filter 59
- texture filters, Texture Fill 59
- threshold 63
- thumbnails 114
  - Mac OS 160
  - Windows 161
- TIFF documents
  - layered 158
  - saving 186
- tool tips 161
- tracking, text 184
- transmission info 98
- trapping 95
- Twirl filter 59
- type units 161

## U

- underlining 184
- units
  - ruler 160
  - type 161
- UnitValue object 31
- Unsharp Mask filter 59
- URLs, document 99
- UTF8 Encoding 112

## V

- version
  - application 46
  - scripting interface 46
- video alpha 161
- video filters
  - De-Interlace 55
  - NTSC 58
- visibility
  - channels 73
  - layer comps 124
  - layers 54

## W

- warp 184
- Wave filter 59
- Web photo galleries
  - See galleries.
- webSnap 107
- Windows
  - filetypes 46
- word spacing 180–183
- work paths
  - designating 197
  - from selected area 167
- wrapping, text 181

**X**

XML 187

xmp metadata 91, 187

**Z**

Zigzag filter 59, 60

zoom 160



