

# **Uitbreiding van ADOBE® FLASH® PROFESSIONAL CS5 en CS5.5**

## **Juridische kennisgeving**

Zie [http://help.adobe.com/nl\\_NL/legalnotices/index.html](http://help.adobe.com/nl_NL/legalnotices/index.html) voor de juridische kennisgeving.

# Inhoud

## Hoofdstuk 1: Inleiding

Werken met de JavaScript-API .....	1
Nieuw in de JavaScript API .....	5
JavaScript-API-objecten .....	7
Voorbeeldimplementaties .....	13

## Hoofdstuk 2: Functies en methoden op hoofdniveau

activate() .....	15
alert() .....	16
configureTool() .....	16
confirm() .....	17
deactivate() .....	18
keyDown() .....	19
keyUp() .....	19
mouseDoubleClick() .....	20
mouseDown() .....	20
mouseMove() .....	21
mouseUp() .....	22
notifySettingsChanged() .....	23
prompt() .....	23
setCursor() .....	24

## Hoofdstuk 3: Object actionsPanel

actionsPanel.getClassForObject() .....	25
actionsPanel.getScriptAssistMode() .....	26
actionsPanel.getSelectedText() .....	27
actionsPanel.getText() .....	27
actionsPanel.hasSelection() .....	28
actionsPanel.replaceSelectedText() .....	28
actionsPanel.setScriptAssistMode() .....	29
actionsPanel.setSelection() .....	30
actionsPanel.setText() .....	31

## Hoofdstuk 4: Object BitmapInstance

bitmapInstance.getBits() .....	32
bitmapInstance.hPixels .....	33
bitmapInstance.setBits() .....	34
bitmapInstance.vPixels .....	34

## Hoofdstuk 5: Object BitmapItem

bitmapItem.allowSmoothing .....	37
bitmapItem.compressionType .....	37
bitmapItem.exportToFile() .....	37
bitmapItem.fileLastModifiedDate .....	38

**Inhoud**

bitmapItem.originalCompressionType .....	39
bitmapItem.quality .....	39
bitmapItem.sourceFileExists .....	40
bitmapItem.sourceFilesCurrent .....	40
bitmapItem.sourceFilePath .....	41
bitmapItem.useDeblocking .....	41
bitmapItem.useImportedJPEGQuality .....	42
 <b>Hoofdstuk 6: Object CompiledClipInstance</b>	
compiledClipInstance.accName .....	43
compiledClipInstance.actionScript .....	44
compiledClipInstance.description .....	44
compiledClipInstance.forceSimple .....	45
compiledClipInstance.shortcut .....	45
compiledClipInstance.silent .....	46
compiledClipInstance.tabIndex .....	46
 <b>Hoofdstuk 7: Object compilerErrors</b>	
compilerErrors.clear() .....	47
compilerErrors.save() .....	48
 <b>Hoofdstuk 8: Object ComponentInstance</b>	
componentInstance.parameters .....	49
 <b>Hoofdstuk 9: Object componentsPanel</b>	
componentsPanel.addItemToDocument() .....	50
componentsPanel.reload() .....	51
 <b>Hoofdstuk 10: Object Contour</b>	
contour.fill .....	52
contour.getHalfEdge() .....	53
contour.interior .....	54
contour.orientation .....	54
 <b>Hoofdstuk 11: Object Document</b>	
document.accName .....	64
document.addDataToDocument() .....	65
document.addDataToSelection() .....	65
document.addFilter() .....	66
document.addItem() .....	67
document.addNewLine() .....	68
document.addNewOval() .....	68
document.addNewPrimitiveOval() .....	69
document.addNewPrimitiveRectangle() .....	70
document.addNewPublishProfile() .....	71
document.addNewRectangle() .....	72
document.addNewScene() .....	73
document.addNewText() .....	73

document.align()	74
document.allowScreens()	75
document.arrange()	76
document.as3AutoDeclare	76
document.as3Dialect	77
document.as3ExportFrame	77
document.as3StrictMode	78
document.as3WarningsMode	78
document.asVersion	79
document.autoLabel	79
document.backgroundColor	80
document.breakApart()	80
document.canEditSymbol()	81
document.canRevert()	81
document.canTestMovie()	82
document.canTestScene()	82
document.changeFilterOrder()	83
document.clipCopy()	84
document.clipCut()	85
document.clipPaste()	85
document.close()	86
document.convertLinesToFills()	86
document.convertToSymbol()	87
document.crop()	88
document.currentPublishProfile	88
document.currentTimeline	89
document.debugMovie()	89
document.deleteEnvelope()	90
document.deletePublishProfile()	90
document.deleteScene()	91
document.deleteSelection()	92
document.description	92
document.disableAllFilters()	93
document.disableFilter()	93
document.disableOtherFilters()	94
document.distribute()	94
document.distributeToLayers()	95
document.docClass	96
document.documentHasData()	96
document.duplicatePublishProfile()	97
document.duplicateScene()	97
document.duplicateSelection()	98
document.editScene()	98
document.enableAllFilters()	99
document.enableFilter()	100
document.enterEditMode()	100

**Inhoud**

document.exitEditMode()	101
document.exportPNG()	102
document.exportPublishProfile()	102
document.exportPublishProfileString()	103
document.exportSWF()	103
document.externalLibraryPath	104
document.forceSimple	105
document.frameRate	105
document.getAlignToDocument()	106
document.getBlendMode()	106
document.getCustomFill()	107
document.getCustomStroke()	108
document.getDataFromDocument()	109
document.getElementProperty()	109
document.getElementTextAttr()	110
document.getFilters()	111
document.getMetadata()	112
document.getMobileSettings()	112
document.getPlayerVersion()	113
document.getSelectionRect()	114
document.getTextString()	115
document.getTimeline()	115
document.getTransformationPoint()	116
document.group()	117
document.height	117
document.id	118
document.importFile()	118
document.importPublishProfile()	119
document.importPublishProfileString()	120
document.importSWF()	120
document.intersect()	121
document.library	121
document.libraryPath	122
document.livePreview	122
document.loadCuepointXML()	123
document.match()	123
document.mouseClick()	124
document.mouseDbIClk()	125
document.moveSelectedBezierPointsBy()	126
document.moveSelectionBy()	126
document.name	127
document.optimizeCurves()	127
document.path	128
document.pathURI	128
document.publish()	129
document.publishProfiles	130

**Inhoud**

document.punch()	130
document.removeAllFilters()	131
document.removeDataFromDocument()	131
document.removeDataFromSelection()	132
document.removeFilter()	132
document.renamePublishProfile()	133
document.renameScene()	133
document.reorderScene()	134
document.resetOvalObject()	135
document.resetRectangleObject()	135
document.resetTransformation()	136
document.revert()	136
document.rotate3DSelection()	137
document.rotateSelection()	137
document.save()	138
document.saveAndCompact()	139
document.scaleSelection()	139
document.screenOutline	140
document.selectAll()	141
document.selection	141
document.selectNone()	143
document.setAlignToDocument()	144
document.setBlendMode()	144
document.setCustomFill()	145
document.setCustomStroke()	146
document.setElementProperty()	146
document.setElementTextAttr()	147
document.setFillColor()	148
document.setFilterProperty()	148
document.setFilters()	149
document.setInstanceAlpha()	150
document.setInstanceBrightness()	150
document.setInstanceTint()	151
document.setMetadata()	152
document.setMobileSettings()	153
document.setOvalObjectProperty()	154
document.setPlayerVersion()	155
document.setRectangleObjectProperty()	155
document.setSelectionBounds()	156
document.setSelectionRect()	157
document.setStageVanishingPoint()	158
document.setStageViewAngle()	158
document.setStroke()	159
document.setStrokeColor()	159
document.setStrokeSize()	160
document.setStrokeStyle()	160

**Inhoud**

document.setTextRectangle()	161
document.setTextSelection()	162
document.setTextString()	162
document.setTransformationPoint()	163
document.silent	164
document.skewSelection()	165
document.smoothSelection()	165
document.sourcePath	166
document.space()	166
document.straightenSelection()	167
document.swapElement()	168
document.swapStrokeAndFill()	168
document.testMovie()	169
document.testScene()	169
document.timelines	170
document.traceBitmap()	170
document.translate3DCenter()	171
document.translate3DSelection()	171
document.transformSelection()	172
document.unGroup()	173
document.union()	173
document.unlockAllElements()	174
document.viewMatrix	174
document.width	175
document.xmlPanel()	175
document.zoomFactor	176
 <b>Hoofdstuk 12: Object drawingLayer</b>	
drawingLayer.beginDraw()	177
drawingLayer.beginFrame()	178
drawingLayer.cubicCurveTo()	179
drawingLayer.curveTo()	179
drawingLayer.drawPath()	180
drawingLayer.endDraw()	180
drawingLayer.endFrame()	181
drawingLayer.lineTo()	181
drawingLayer.moveTo()	182
drawingLayer.newPath()	182
drawingLayer.setColor()	183
drawingLayer.setFill()	184
drawingLayer.setStroke()	184
 <b>Hoofdstuk 13: Object Edge</b>	
edge.cubicSegmentIndex	185
edge.getControl()	186
edge.getHalfEdge()	186
edge.id	187



**Inhoud**

edge.isLine .....	187
edge.setControl() .....	188
edge.splitEdge() .....	188
edge.stroke .....	189

**Hoofdstuk 14: Object Element**

element.depth .....	191
element.elementType .....	192
element.getPersistentData() .....	192
element.getTransformationPoint() .....	193
element.hasPersistentData() .....	194
element.height .....	194
element.layer .....	194
element.left .....	195
element.locked .....	195
element.matrix .....	196
element.name .....	196
element.removePersistentData() .....	197
element.rotation .....	197
element.scaleX .....	198
element.scaleY .....	198
element.selected .....	199
element.setPersistentData() .....	199
element.setTransformationPoint() .....	200
element.skewX .....	201
element.skewY .....	201
element.top .....	202
element.transformX .....	202
element.transformY .....	203
element.width .....	203
element.x .....	203
element.y .....	204

**Hoofdstuk 15: Object Fill**

fill.bitmapIsClipped .....	205
fill.bitmapPath .....	206
fill.color .....	206
fill.colorArray .....	207
fill.focalPoint .....	207
fill.linearRGB .....	208
fill.matrix .....	208
fill.overflow .....	209
fill.posArray .....	210
fill.style .....	210

**Inhoud****Hoofdstuk 16: Object Filter**

filter.angle	212
filter.blurX	212
filter.blurY	213
filter.brightness	213
filter.color	214
filter.contrast	215
filter.distance	215
filter.enabled	216
filter.hideObject	216
filter.highlightColor	217
filter.hue	217
filter.inner	218
filter.knockout	219
filter.name	219
filter.quality	220
filter.saturation	220
filter.shadowColor	221
filter.strength	222
filter.type	222

**Hoofdstuk 17: Object flash (fl)**

fl.actionsPanel	227
fl.addEventListener()	227
fl.as3PackagePaths	228
fl.browseForFileURL()	229
fl.browseForFolderURL()	230
fl.clearPublishCache()	230
fl.clipCopyString()	231
fl.closeAll()	231
fl.closeAllPlayerDocuments()	232
fl.closeDocument()	233
fl.compilerErrors	233
fl.componentsPanel	234
fl.configDirectory	234
fl.configURI	234
fl.contactSensitiveSelection	235
fl.createDocument()	235
fl.createNewDocList	236
fl.createNewDocListType	236
fl.createNewTemplateList	237
fl.documents	237
fl.drawingLayer	238
fl.exportPublishProfileString()	238
fl.externalLibraryPath	239
fl.fileExists()	239
fl.findDocumentDOM()	240

**Inhoud**

fl.findDocumentIndex()	240
fl.findObjectInDocByName()	241
fl.findObjectInDocByType()	242
fl.flexSDKPath	244
fl.getAppMemoryInfo()	244
fl.getDocumentDOM()	245
fl.getSwfPanel()	246
fl.installedPlayers	246
fl.isFontInstalled()	247
fl.languageCode	248
fl.libraryPath	248
fl.mapPlayerURL()	249
fl.Math	249
fl.mruRecentFileList	250
fl.mruRecentFileListType	250
fl.objectDrawingMode	251
fl.openDocument()	251
fl.openScript()	252
fl.outputPanel	253
fl.packagePaths	253
fl.presetPanel	253
fl.publishCacheDiskSizeMax	254
fl.publishCacheEnabled	254
fl.publishCacheMemoryEntrySizeLimit	255
fl.publishCacheMemorySizeMax	255
fl.publishDocument()	256
fl.quit()	256
fl.reloadEffects()	257
fl.reloadTools()	258
fl.removeEventListener()	258
fl.resetAS3PackagePaths()	259
fl.resetPackagePaths()	259
fl.revertDocument()	260
fl.runScript()	260
fl.saveAll()	261
fl.saveDocument()	262
fl.saveDocumentAs()	263
fl.scriptURI	263
fl.selectElement()	264
fl.selectTool()	265
fl.setActiveWindow()	266
fl.showIdleMessage()	266
fl.sourcePath	267
fl.swfPanels	268
fl.toggleBreakpoint()	268
fl.tools	269

**Inhoud**

fl.trace()	269
fl.version	270
fl.xmlui	270

**Hoofdstuk 18: Object FLfile**

FLfile.copy()	272
FLfile.createFolder()	273
FLfile.exists()	274
FLfile.getAttributes()	275
FLfile.getCreationDate()	276
FLfile.getCreationDateObj()	276
FLfile.getModificationDate()	277
FLfile.getModificationDateObj()	278
FLfile.getSize()	279
FLfile.listFolder()	279
FLfile.platformPathToURI()	280
FLfile.read()	281
FLfile.remove()	282
FLfile.setAttributes()	283
FLfile.uriToPlatformPath()	284
FLfile.write()	285

**Hoofdstuk 19: Object folderItem****Hoofdstuk 20: Object fontItem**

fontItem.bitmap	287
fontItem.bold	288
fontItem.embeddedCharacters	288
fontItem.embedRanges	289
fontItem.embedVariantGlyphs	289
fontItem.font	291
fontItem.isDefineFont4Symbol	291
fontItem.italic	292
fontItem.size	292

**Hoofdstuk 21: Object Frame**

frame.convertMotionObjectTo2D()	295
frame.convertMotionObjectTo3D()	295
frame.actionScript	296
frame.duration	296
frame.elements	297
frame.getCustomEase()	297
frame.getMotionObjectXML()	298
frame.hasCustomEase	298
frame.hasMotionPath()	299
frame.is3DMotionObject()	299
frame.isMotionObject()	300
frame.labelType	301

**Inhoud**

frame.motionTweenOrientToPath	301
frame.motionTweenRotate	301
frame.motionTweenRotateTimes	302
frame.motionTweenScale	302
frame.motionTweenSnap	303
frame.motionTweenSync	303
frame.name	303
frame.selectMotionPath()	304
frame.setCustomEase()	305
frame.setMotionObjectDuration()	305
frame.setMotionObjectXML()	306
frame.shapeTweenBlend	307
frame.soundEffect	307
frame.soundLibraryItem	307
frame.soundLoop	308
frame.soundLoopMode	308
frame.soundName	309
frame.soundSync	309
frame.startFrame	309
frame.tweenEasing	310
frame.tweenInstanceName	310
frame.tweenType	311
frame.useSingleEaseCurve	311
<b>Hoofdstuk 22: Object HalfEdge</b>	
halfEdge.getEdge()	312
halfEdge.getNext()	313
halfEdge.getOppositeHalfEdge()	313
halfEdge.getPrev()	314
halfEdge.getVertex()	315
halfEdge.id	315
halfEdge.index	316
<b>Hoofdstuk 23: Object Instance</b>	
instance.instanceType	317
instance.libraryItem	318
<b>Hoofdstuk 24: Object Item</b>	
item.addData()	320
item.getData()	320
item.hasData()	321
item.itemType	321
item.linkageBaseClass	322
item.linkageClassName	322
item.linkageExportForAS	323
item.linkageExportForRS	323
item.linkageExportInFirstFrame	324

**Inhoud**

item.linkageIdentifier .....	324
item.linkageImportForRS .....	325
item.linkageURL .....	325
item.name .....	326
item.removeData() .....	326

**Hoofdstuk 25: Object Layer**

layer.color .....	327
layer.frameCount .....	328
layer.frames .....	328
layer.height .....	329
layer.layerType .....	329
layer.locked .....	330
layer.name .....	330
layer.outline .....	330
layer.parentLayer .....	331
layer.visible .....	331

**Hoofdstuk 26: Object library**

library.addItemToDocument() .....	334
library.addNewItem() .....	335
library.deleteItem() .....	335
library.duplicateItem() .....	336
library.editItem() .....	336
library.expandFolder() .....	337
library.findItemIndex() .....	338
library.getItemProperty() .....	338
library.getItemType() .....	339
library.getSelectedItems() .....	339
library.importEmbeddedSWF() .....	340
library.itemExists() .....	341
library.items .....	341
library.moveToFolder() .....	341
library.newFolder() .....	342
library.renameItem() .....	343
library.selectAll() .....	343
library.selectItem() .....	344
library.selectNone() .....	345
library.setItemProperty() .....	345
library.updateItem() .....	346

**Hoofdstuk 27: Object Math**

Math.concatMatrix() .....	347
Math.invertMatrix() .....	348
Math.pointDistance() .....	348

**Inhoud****Hoofdstuk 28: Object Matrix**

matrix.a	350
matrix.b	351
matrix.c	351
matrix.d	352
matrix.tx	352
matrix.ty	353

**Hoofdstuk 29: Object outputPanel**

outputPanel.clear()	354
outputPanel.save()	355
outputPanel.trace()	355

**Hoofdstuk 30: Object Oval**

OvalObject.closePath	357
OvalObject.endAngle	358
OvalObject.innerRadius	358
OvalObject.startAngle	359

**Hoofdstuk 31: Object Parameter**

parameter.category	360
parameter.insertItem()	361
parameter.listIndex	361
parameter.name	362
parameter.removeItem()	362
parameter.value	363
parameter.valueType	364
parameter.verbose	364

**Hoofdstuk 32: Object Path**

path.addCubicCurve()	365
path.addCurve()	366
path.addPoint()	367
path.clear()	367
path.close()	368
path.makeShape()	368
path.newContour()	369
path.nPts	370

**Hoofdstuk 33: Object presetItem**

presetItem.isDefault	371
presetItem.isFolder	372
presetItem.level	372
presetItem.name	373
presetItem.open	373
presetItem.path	374

**Inhoud****Hoofdstuk 34: Object presetPanel**

presetPanel.addNewItem()	376
presetPanel.applyPreset()	376
presetPanel.deleteFolder()	377
presetPanel.deleteItem()	378
presetPanel.expandFolder()	378
presetPanel.exportItem()	379
presetPanel.findItemIndex()	380
presetPanel.getSelectedItems()	381
presetPanel.importItem()	382
presetPanel.items	382
presetPanel.moveToFolder()	383
presetPanel.newFolder()	384
presetPanel.renameItem()	384
presetPanel.selectItem()	385

**Hoofdstuk 35: Object Rectangle**

RectangleObject.bottomLeftRadius	386
RectangleObject.bottomRightRadius	387
RectangleObject.lockFlag	387
RectangleObject.topLeftRadius	388
RectangleObject.topRightRadius	388

**Hoofdstuk 36: Object Shape**

shape.beginEdit()	390
shape.contours	390
shape.deleteEdge()	391
shape.edges	391
shape.endEdit()	391
shape.getCubicSegmentPoints()	392
shape.isDrawingObject	393
shape.isGroup	393
shape.isOvalObject	394
shape.isRectangleObject	394
shape.members	395
shape.numCubicSegments	395
shape.vertices	396

**Hoofdstuk 37: Object SoundItem**

soundItem.bitRate	398
soundItem.bits	398
soundItem.compressionType	399
soundItem.convertStereoToMono	399
soundItem.exportToFile()	400
soundItem.fileLastModifiedDate	401
soundItem.originalCompressionType	401
soundItem.quality	402



**Inhoud**

soundItem.sampleRate .....	402
soundItem.sourceFileExists .....	403
soundItem.sourceFilesCurrent .....	403
soundItem.sourceFilePath .....	404
soundItem.useImportedMP3Quality .....	404

**Hoofdstuk 38: Object Stroke**

stroke.breakAtCorners .....	406
stroke.capType .....	406
stroke.color .....	407
stroke.curve .....	407
stroke.dash1 .....	408
stroke.dash2 .....	408
stroke.density .....	409
stroke.dotSize .....	409
stroke.dotSpace .....	410
stroke.hatchThickness .....	410
stroke.jiggle .....	411
stroke.joinType .....	411
stroke.length .....	411
stroke.miterLimit .....	412
stroke.pattern .....	412
stroke.rotate .....	413
stroke.scaleType .....	413
stroke.shapeFill .....	414
stroke.space .....	414
stroke.strokeHinting .....	415
stroke.style .....	415
stroke.thickness .....	416
stroke.variation .....	416
stroke.waveHeight .....	417
stroke.waveLength .....	417

**Hoofdstuk 39: Object swfPanel**

swfPanel.call() .....	419
swfPanel.name .....	421
swfPanel.path .....	422
swfPanel.setFocus() .....	422

**Hoofdstuk 40: Object SymbolInstance**

symbolInstance.accName .....	425
symbolInstance.actionScript .....	426
symbolInstance.backgroundColor .....	426
symbolInstance.bitmapRenderMode .....	427
symbolInstance.blendMode .....	427
symbolInstance.buttonTracking .....	428
symbolInstance.cacheAsBitmap .....	428

**Inhoud**

symbolInstance.colorAlphaAmount	429
symbolInstance.colorAlphaPercent	429
symbolInstance.colorBlueAmount	430
symbolInstance.colorBluePercent	430
symbolInstance.colorGreenAmount	430
symbolInstance.colorGreenPercent	431
symbolInstance.colorMode	431
symbolInstance.colorRedAmount	431
symbolInstance.colorRedPercent	432
symbolInstance.description	432
symbolInstance.filters	433
symbolInstance.firstFrame	433
symbolInstance.forceSimple	434
symbolInstance.loop	434
symbolInstance.shortcut	435
symbolInstance.silent	435
symbolInstance.symbolType	436
symbolInstance.tabIndex	436
symbolInstance.usesBackgroundColor	436
symbolInstance.visible	437
<b>Hoofdstuk 41: Object SymbolItem</b>	
symbolItem.convertToCompiledClip()	438
symbolItem.exportSWC()	439
symbolItem.exportSWF()	440
symbolItem.scalingGrid	440
symbolItem.scalingGridRect	441
symbolItem.sourceAutoUpdate	441
symbolItem.sourceFilePath	441
symbolItem.sourceLibraryName	442
symbolItem.symbolType	442
symbolItem.timeline	443
<b>Hoofdstuk 42: Object Text</b>	
text.accName	445
text.antiAliasSharpness	446
text.antiAliasThickness	446
text.autoExpand	447
text.border	447
text.description	448
text.embeddedCharacters	448
text.embedRanges	449
text.embedVariantGlyphs	449
text.fontRenderingMode	450
text.getTextAttr()	451
text.getTextString()	452
text.length	452

**Inhoud**

text.lineType .....	453
text.maxCharacters .....	453
text.orientation .....	454
text.renderAsHTML .....	454
text.scrollable .....	454
text.selectable .....	455
text.selectionEnd .....	455
text.selectionStart .....	456
text.setTextAttr() .....	456
text.setTextString() .....	457
text.shortcut .....	458
text.silent .....	459
text.tabIndex .....	459
text.textRuns .....	459
text.textType .....	460
text.useDeviceFonts .....	460
text.variableName .....	461
 <b>Hoofdstuk 43: Object TextAttrs</b>	
textAttrs.aliasText .....	463
textAttrs.alignment .....	463
textAttrs.autoKern .....	463
textAttrs.bold .....	464
textAttrs.characterPosition .....	464
textAttrs.characterSpacing .....	465
textAttrs.face .....	465
textAttrs.fillColor .....	465
textAttrs.indent .....	466
textAttrs.italic .....	466
textAttrs.leftMargin .....	467
textAttrs.letterSpacing .....	467
textAttrs.lineSpacing .....	467
textAttrs.rightMargin .....	468
textAttrs.rotation .....	468
textAttrs.size .....	469
textAttrs.target .....	469
textAttrs.url .....	469
 <b>Hoofdstuk 44: Object TextRun</b>	
textRun.textAttrs .....	471
textRun.characters .....	471
 <b>Hoofdstuk 45: Object Timeline</b>	
timeline.addMotionGuide() .....	475
timeline.addNewLayer() .....	476
timeline.clearFrames() .....	477
timeline.clearKeyframes() .....	477

**Inhoud**

timeline.convertToBlankKeyframes()	478
timeline.convertToKeyframes()	479
timeline.copyFrames()	480
timeline.copyLayers()	480
timeline.copyMotion()	481
timeline.copyMotionAsAS3()	482
timeline.createMotionObject()	482
timeline.createMotionTween()	483
timeline.currentFrame	484
timeline.currentLayer	484
timeline.cutFrames()	484
timeline.cutLayers()	485
timeline.deleteLayer()	486
timeline duplicateLayers()	486
timeline.expandFolder()	487
timeline.findLayerIndex()	488
timeline.frameCount	489
timeline.getFrameProperty()	489
timeline.getGuidelines()	490
timeline.getLayerProperty()	490
timeline.getSelectedFrames()	491
timeline.getSelectedLayers()	492
timeline.insertBlankKeyframe()	492
timeline.insertFrames()	493
timeline.insertKeyframe()	494
timeline.layerCount	495
timeline.layers	495
timeline.libraryItem	496
timeline.name	496
timeline.pasteFrames()	497
timeline.pasteLayers()	497
timeline.pasteMotion()	498
timeline.removeFrames()	499
timeline.removeMotionObject()	499
timeline.reorderLayer()	500
timeline.reverseFrames()	501
timeline.selectAllFrames()	501
timeline setFrameProperty()	502
timeline.setGuidelines()	503
timeline.setLayerProperty()	503
timeline.setSelectedFrames()	504
timeline.setSelectedLayers()	505
timeline.showLayerMasking()	506
timeline.startPlayback()	506
timeline.stopPlayback()	507

**Inhoud****Hoofdstuk 46: Object ToolObj**

toolObj.depth	509
toolObj.enablePIControl()	509
toolObj.iconID	510
toolObj.position	511
toolObj.setIcon()	511
toolObj.setMenuString()	512
toolObj.setOptionsFile()	512
toolObj.setPI()	513
toolObj.setToolName()	514
toolObj.setToolTip()	515
toolObj.showPIControl()	515
toolObj.showTransformHandles()	516

**Hoofdstuk 47: Object Tools**

tools.activeTool	519
tools.altIsDown	519
tools.constrainPoint()	519
tools.ctrlIsDown	520
tools.getKeyDown()	520
tools.mouselsDown	521
tools.penDownLoc	521
tools.penLoc	522
tools.setCreatingBbox()	522
tools.setCursor()	523
tools.shiftIsDown	523
tools.snapPoint()	524
tools.toolObjs	524

**Hoofdstuk 48: Object Vertex**

vertex.getHalfEdge()	525
vertex.setLocation()	526
vertex.x	526
vertex.y	527

**Hoofdstuk 49: Object VideoItem**

videoItem.exportToFLV()	528
videoItem.fileLastModifiedDate	529
videoItem.sourceFileExists	529
videoItem.sourceFilesCurrent	530
videoItem.sourceFilePath	530
videoItem.videoType	531

**Hoofdstuk 50: Object XMLUI**

xmlui.accept()	533
xmlui.cancel()	533
xmlui.get()	534
xmlui.getControlItemElement()	534

**Inhoud**

xmlui.setEnabled()	535
xmlui.getVisible()	535
xmlui.set()	536
xmlui.setControlItemElement()	537
xmlui.setControlItemElements()	537
xmlui.setEnabled()	538
xmlui.setVisible()	539

**Hoofdstuk 51: Uitbreidbaarheid in C**

Over uitbreidbaarheid	540
C-functies integreren	540
Gegevenstypen	546
De C-API	546

# Hoofdstuk 1: Inleiding

Als gebruiker van Adobe® Flash® Professional CS5 of CS5.5 bent u waarschijnlijk al bekend met Adobe® ActionScript®. Hiermee kunt u scripts maken die in Adobe® Flash® Player worden uitgevoerd. De Flash JavaScript-API (programmeerinterface voor toepassingen) die in dit document wordt beschreven is een aanvullend programmeerhulpmiddel waarin u scripts kunt maken die in de ontwerpomgeving worden uitgevoerd.

In dit document treft u de beschrijving aan van de objecten, methoden en eigenschappen die beschikbaar zijn in de JavaScript-API. Er wordt aangenomen dat u bekend bent met het gebruik van de gedocumenteerde opdrachten wanneer u in de ontwerpomgeving werkt. Als u een vraag hebt over de werking van een bepaalde opdracht, kunt u gebruikmaken van andere documenten in Flash Help (zoals *Flash gebruiken*) om die informatie op te zoeken.

In dit document wordt ook aangenomen dat u bekend bent met JavaScript- of ActionScript-syntaxis en met standaardprogrammeerconcepten zoals functies, parameters en gegevenstypen.

## Werken met de JavaScript-API

Met de Flash JavaScript-API kunt u scripts schrijven om verschillende acties uit te voeren in de ontwerpomgeving van Flash (terwijl het programma Flash door een gebruiker is geopend). Deze functionaliteit is anders dan de ActionScript-taal, waarmee u scripts kunt schrijven om acties uit te voeren in de Flash Player-omgeving (terwijl een SWF-bestand wordt afgespeeld). Deze functionaliteit is ook anders dan JavaScript-opdrachten, die u kunt gebruiken op pagina's die in een webbrowser worden weergegeven.

Met de JavaScript-API kunt u Flash-toepassingscripts schrijven om het ontwerpproces mede te stroomlijnen. U kunt bijvoorbeeld scripts schrijven om routinetaken te automatiseren of om aangepaste gereedschappen aan het deelvenster Gereedschappen toe te voegen.

De Flash JavaScript-API is gemaakt om te lijken op de Adobe® Dreamweaver® en Adobe® Fireworks® JavaScript-API (die zijn gemaakt op basis van de Netscape JavaScript-API). De Flash JavaScript-API is gebaseerd op een Document Object Model (DOM), waarmee Flash-documenten met behulp van JavaScript-objecten kunnen worden benaderd. De Flash JavaScript-API bevat alle elementen van de Netscape JavaScript-API, plus de Flash DOM. Deze extra objecten en de bijbehorende methoden en eigenschappen worden in dit document beschreven. U kunt alle elementen van de oorspronkelijke JavaScript-taal in een Flash-script gebruiken, maar alleen de elementen die in de context van een Flash-document zinnig zijn, hebben een effect.

De JavaScript-API bevat ook methoden waarmee u uitbreidbaarheid kunt implementeren met behulp van een combinatie van JavaScript en eigen code in C. Zie “[Uitbreidbaarheid in C](#)” op pagina 540 voor meer informatie.

De JavaScript-interpret in Flash is de Mozilla SpiderMonkey-engine, versie 1.6, die op het web beschikbaar is op [www.mozilla.org/js/spidermonkey/](http://www.mozilla.org/js/spidermonkey/). SpiderMonkey is een van de twee referentie-implementaties van de JavaScript-taal die door Mozilla.org is ontwikkeld. Dit is dezelfde engine die in de browser Mozilla is ingesloten.

SpiderMonkey implementeert de kern-JavaScript-taal, zoals gedefinieerd in ECMAScript (ECMA-262) Edition 3 Language Specification en voldoet volledig aan de specificatie. Alleen de browserspecifieke hostobjecten die geen deel uitmaken van de specificatie ECMA-262, worden niet ondersteund. Zo wordt in veel JavaScript-naslaggidsen onderscheid gemaakt tussen kern-JavaScript en client-JavaScript (browsergerelateerd). Alleen kern-JavaScript is van toepassing op de Flash JavaScript-interpret.

## JSFL-bestanden maken

U kunt Adobe Flash Professional of de teksteditor van uw voorkeur gebruiken om Flash JavaScript-bestanden (.JSFL) te schrijven en te bewerken. Als u Flash gebruikt, hebben deze bestanden standaard de extensie .jsfl. Als u een script wilt schrijven, selecteert u Bestand > Nieuw > Flash JavaScript-bestand.

U kunt ook een JSFL-bestand maken door opdrachten in het deelvenster Historie te selecteren. Klik vervolgens op de knop Opslaan in het deelvenster Historie of selecteer de opdracht Opslaan als in het deelvenstermenu. Het opdrachtbestand (.JSFL) wordt opgeslagen in de map met opdrachten (zie “[JSFL-bestanden opslaan](#)” op pagina 2). U kunt het bestand vervolgens op dezelfde manier openen en bewerken als elk scriptbestand.

Het deelvenster Historie bevat nog enkele nuttige opties. U kunt geselecteerde opdrachten naar het Klembord kopiëren en u kunt JavaScript-opdrachten weergeven die zijn gegenereerd terwijl u in Flash werkt.

### Opdrachten van het deelvenster Historie naar het klembord kopiëren:

- 1 Selecteer een of meer opdrachten in het deelvenster Historie.
- 2 Ga als volgt te werk:
  - Klik op de knop Kopiëren.
  - Selecteer Stappen kopiëren in het deelvenstermenu.

### JavaScript-opdrachten weergeven in het deelvenster Historie:

- Selecteer Weergave > JavaScript in deelvenster in het deelvenstermenu.

## JSFL-bestanden opslaan

U kunt JSFL-scripts beschikbaar houden in de ontwerpomgeving van Flash door deze in een van de mappen in de configuratiemap op te slaan. Standaard kunt u de configuratiemap hier vinden:

- Windows® 7™:  
*opstartstation\Users\gebruikersnaam\AppData\Local\Adobe\Flash CS5 of CS5.5\taal\Configuration\*
- Windows® Vista™:  
*opstartstation\Users\gebruikersnaam\Local Settings\Application Data\Adobe\Flash CS5 of CS5.5\taal\Configuration\*
- Windows XP:  
*opstartstation\Documents and Settings\gebruikersnaam\Local Settings\Application Data\Adobe\Flash CS5 of CS5.5\taal\Configuration\*
- Mac OS® X:  
*Macintosh HD/Users/gebruikersnaam/Library/Application Support/Adobe/Flash CS5 of CS5.5/taal/Configuration/*

Als u de configuratiemap wilt zoeken, gebruikt u `fl.configDirectory` of `fl.configURI`, zoals in het volgende voorbeeld:

```
// store directory to a variable
var configDir = fl.configDirectory;
// display directory in the Output panel
fl.trace(fl.configDirectory);
```



In de configuratiemap kunnen de volgende mappen scripts bevatten die u in de ontwerpomgeving kunt gebruiken: Behaviors (om gedrag in de gebruikersinterface te ondersteunen); Commands (voor scripts die in het menu Opdrachten verschijnen); JavaScript (voor scripts die door Script Assist worden gebruikt om de besturingselementen voor gebruikersinterfaces te vullen); Tools (voor uitbreidbare gereedschappen in het deelvenster Gereedschappen) en WindowSWF (voor deelvensters die in het menu Windows worden weergegeven). In dit document worden de scripts beschreven die worden gebruikt voor opdrachten en gereedschappen.

Als u een script in de map Commands bewerkt, is het nieuwe script direct beschikbaar in Flash. Als u een script voor een uitbreidbaar gereedschap bewerkt, sluit u Flash en start u het opnieuw, of gebruikt u de opdracht `fl.reloadTools()`. Als u echter een script hebt gebruikt om een uitbreidbaar gereedschap aan het deelvenster Gereedschappen toe te voegen en u dat script vervolgens bewerkt, moet u het gereedschap van het deelvenster verwijderen en weer toevoegen, of Flash afsluiten en opnieuw opstarten. Het gewijzigde gereedschap is dan beschikbaar.

U kunt opdracht- en gereedschapsbestanden op twee locaties opslaan, zodat deze in de ontwerpomgeving te gebruiken zijn.

- Voor scripts die als opties in het menu Opdrachten verschijnen, moet u het JSFL-bestand opslaan in de opdrachtenmap op de volgende locatie:

Besturingssysteem	Locatie
Windows 7	<i>opstartstation\Users\gebruikersnaam\AppData\Local\Adobe\Flash CS5 of CS5.5\taal\Configuration\Commands</i>
Windows Vista	<i>opstartstation\Gebruikers\gebruikersnaam\Local Settings\Application Data\Adobe\Flash CS5\taal\Configuration\Commands</i>
Windows XP	<i>opstartstation\Documents and Settings\gebruiker\Local Settings\Application Data\Adobe\Flash CS5\taal\Configuration\Commands</i>
Mac OS X	<i>Macintosh HD/Users/gebruikersnaam/Library/Application Support/Adobe/Flash CS5/taal/Configuration/Commands</i>

- Voor scripts die als uitbreidbare gereedschappen in het deelvenster Gereedschappen verschijnen, moet u het JSFL-bestand opslaan in de map Tools op de volgende locatie:

Besturingssysteem	Locatie
Windows 7	<i>opstartstation\Users\gebruikersnaam\AppData\Local\Adobe\Flash CS5 of CS5.5\taal\Configuration\Tools</i>
Windows Vista	<i>opstartstation\Gebruikers\gebruikersnaam\Local Settings\Application Data\Adobe\Flash CS5\taal\Configuration\Tools</i>
Windows XP	<i>opstartstation\Documents and Settings\gebruiker\Local Settings\Application Data\Adobe\Flash CS5\taal\Configuration\Tools</i>
Mac OS X	<i>Macintosh HD/Users/gebruikersnaam/Library/Application Support/Adobe/Flash CS5/taal/Configuration/Tools</i>

Als een JSFL-bestand andere bestanden heeft die erbij horen, zoals XML-bestanden, slaat u deze in dezelfde map op als het JSFL-bestand.

## Scripts uitvoeren

U kunt scripts op verschillende manieren uitvoeren. In dit gedeelte worden de meest gebruikte manieren beschreven.

**Een script uitvoeren dat u nu bekijkt of bewerkt:**

- Klik met de rechtermuisknop (de knop Command op de Macintosh) en kies Script uitvoeren.
- Klik op het pictogram Script uitvoeren op de werkbalk van het scriptvenster.

Met deze optie kunt u een script uitvoeren voordat u het hebt opgeslagen. Met deze optie kunt u ook een script uitvoeren als er geen FLA-bestanden geopend zijn.

**Ga als volgt te werk om een script uit te voeren dat in de opdrachtenmap is:**

- Selecteer in de ontwerpomgeving Opdrachten > *Scriptnaam*.
- Gebruik een sneltoets die u aan het script hebt toegewezen. Als u een sneltoets wilt toewijzen, gebruikt u Bewerken > Sneltoetsen en selecteert u Opdrachten menu Teken in het pop-upmenu Opdrachten. Vouw het knooppunt Commands in de menustructuur uit om een lijst met beschikbare scripts te zien.

**Ga als volgt te werk om een opdrachscript uit te voeren dat niet in de opdrachtenmap is:**

- Selecteer in de ontwerpomgeving Opdrachten > Opdracht uitvoeren en selecteer het script dat u wilt uitvoeren.
- Gebruik vanuit het script de opdracht `fl.runScript()`.
- Dubbelklik in het bestandssysteem op het scriptbestand.

**Een gereedschap aan het deelvenster Gereedschappen toevoegen dat is geïmplementeerd in een JSFL-bestand:**

- 1 Kopieer het JSFL-bestand voor de tool en eventuele andere bijbehorende bestanden naar de map Tools (zie “[JSFL-bestanden opslaan](#)” op pagina 2).
- 2 Selecteer Bewerken > Deelvenster Gereedschappen aanpassen (Windows) of Flash > Deelvenster Gereedschappen aanpassen (Macintosh).
- 3 Voeg het gereedschap toe aan de lijst met beschikbare gereedschappen.
- 4 Klik op OK.

U kunt afzonderlijke JavaScript API-opdrachten aan ActionScript-bestanden toevoegen met behulp van de functie `MMEExecute()`, die wordt beschreven in de *Naslaggids voor ActionScript 3.0 en ActionScript 3.0-componenten*. De functie `MMEExecute()` heeft echter alleen effect wanneer deze wordt gebruikt in de context van een aangepast element van de gebruikersinterface, zoals een component Eigenschapcontrole of een SWF-deelvenster in de ontwerpomgeving. JavaScript API-opdrachten hebben geen effect in Flash Player of buiten de ontwerpomgeving, ook niet wanneer deze vanuit ActionScript worden aangeroepen.

**Een opdracht geven vanuit een ActionScript-script:**

- Gebruik de volgende syntaxis (u kunt verschillende opdrachten in één tekenreeks samenvoegen):

```
MMEExecute(Javascript command string);
```

U kunt een script ook uitvoeren vanaf de opdrachtregel.

**Een script uitvoeren vanaf de opdrachtregel in Windows:**

- Gebruik de volgende syntaxis (voeg waar nodig padinformatie toe):

```
"flash.exe" myTestFile.jsfl
```

**Een script uitvoeren vanuit de “Terminal”-toepassing op de Macintosh:**

- Gebruik de volgende syntaxis (voeg waar nodig padinformatie toe):

```
osascript -e 'tell application "flash" to open alias "Mac OS X:Users:user:myTestFile.jsfl" '
```

De opdracht `osascript` kan ook AppleScript uitvoeren in een bestand. U kunt bijvoorbeeld de volgende tekst in een bestand met de naam `myScript` opnemen:

```
tell application "flash"
open alias "Mac OS X:Users:user:myTestFile.jsfl"
end tell
```

Vervolgens kunt u deze opdracht gebruiken om het script uit te voeren:

```
osascript myScript
```

## Nieuw in de JavaScript API

In Flash CS5 en CS5.5 zijn enkele objecten, methoden en eigenschappen toegevoegd en andere verwijderd. Hieronder vindt u een overzicht van de wijzigingen.

Als u de JavaScript-API nog nooit hebt gebruikt, kunt u deze sectie overslaan en direct naar [“JavaScript-API-objecten”](#) op pagina 7 gaan.

### Nieuwe methoden en eigenschappen

De volgende methoden en eigenschappen voor bestaande objecten zijn nieuw in Flash Pro CS5:

- [Object Document](#)
  - `document.debugMovie()`
  - `document.loadCuepointXML()`
- [Object flash \(fl\)](#)
  - `fl.languageCode`
  - `fl.toggleBreakpoint`
- [Object Frame](#)
  - `frame.convertMotionObjectTo2D()`
  - `frame.convertMotionObjectTo3D()`
  - `frame.getMotionObjectXML()`
  - `frame.hasMotionPath()`
  - `frame.isMotionObject()`
  - `frame.is3DMotionObject()`
  - `frame.selectMotionPath()`
  - `frame.setMotionObjectDuration()`
  - `frame.setMotionObjectXML()`
  - `frame.tweenInstanceName`
- [Object Timeline](#)
  - `timeline.createMotionObject()`
  - `timeline.libraryItem`

- `timeline.removeMotionObject()`
- `timeline.startPlayback`
- `timeline.stopPlayback`

De volgende methoden en eigenschappen voor bestaande objecten zijn nieuw in Flash Pro CS5.5:

- **Object SymbolInstance**
  - `symbolInstance.bitmapRenderMode`
  - `symbolInstance.backgroundColor`
  - `symbolInstance.usesBackgroundColor`
  - `symbolInstance.visible`
- **Object Timeline**
  - `timeline.copyLayers()`
  - `timeline.cutLayers()`
  - `timeline.duplicateLayers()`
  - `timeline.pasteLayers()`
- **Object flash (fl)**
  - `fl.getSwfPanel()`
  - `fl.installedPlayers()`
  - `fl.publishCacheEnabled`
  - `fl.publishCacheDiskSizeMax`
  - `fl.publishCacheMemorySizeMax`
  - `fl.publishCacheMemoryEntrySizeLimit`
  - `fl.clearPublishCache()`
- **Object swfPanel**
  - `swfPanel.setFocus()`

## Andere wijzigingen

De volgende methoden en eigenschappen zijn bijgewerkt in Flash CS5:

- `fl.openScript()`
- `fl.publishDocument()`
- `fontItem.embedRanges`
- `fontItem.embeddedCharacters`
- `fontItem.embedVariantGlyphs`

De volgende objecten en methoden zijn niet meer beschikbaar in Flash CS5:

- Object `Screen`
- Object `ScreenOutline`
- `document.canSaveAVersion()`
- `document.revertToLastVersion()`

- `document.saveAVersion()`
- `document.synchronizeWithHeadVersion()`
- `fl.downloadLatestVersion()`
- `fl.revertDocumentToLastVersion()`
- `fl.saveAVersionOfDocument()`
- `fl.synchronizeDocumentWithHeadVersion()`

## JavaScript-API-objecten

Deze sectie bevat een overzicht van de objecten die beschikbaar zijn in de Flash JavaScript-API en hoe u ermee aan de slag gaat. Alle standaard JavaScript-opdrachten zijn ook beschikbaar wanneer u met de JavaScript-API werkt.

In de volgende tabel vindt u een beknopte beschrijving van de objecten in de JavaScript-API. De objecten staan op alfabetische volgorde.

Object	Beschrijving
Object actionsPanel	Het object actionsPanel vertegenwoordigt het actieve deelvenster Handelingen.
Object BitmapInstance	Het object BitmapInstance is een subklasse van het object Instance en vertegenwoordigt een bitmap in een frame.
Object BitmapItem	Een object BitmapItem verwijst naar een bitmap in de bibliotheek van een document. Het object BitmapItem is een subklasse van het object Item.
Object CompiledClipInstance	Het object CompiledClipInstance is een subklasse van het object Instance.
Object compilerErrors	Het object compilerErrors vertegenwoordigt het deelvenster Compilerfouten. Het is een eigenschap van het object flash ( <code>fl.compilerErrors</code> ).
Object ComponentInstance	Het object ComponentInstance is een subklasse van het object SymbolInstance en vertegenwoordigt een bitmap in een frame.
Object componentsPanel	Het object componentsPanel, dat het deelvenster Componenten vertegenwoordigt, is een eigenschap van het object flash ( <code>fl.componentsPanel</code> ).
Object Contour	Een object Contour vertegenwoordigt een gesloten pad van halve randen op de omtrek van een vorm.
Object Document	Het object Document vertegenwoordigt het werkgebied.
Object drawingLayer	Het object drawingLayer is toegankelijk vanuit JavaScript als onderliggend element van het object flash.
Object Edge	Het object Edge vertegenwoordigt een rand van een vorm in het werkgebied.
Object Element	Alles wat op het werkgebied wordt weergegeven, is van het type Element.
Object Fill	Het object Fill bevat alle eigenschappen van de vulkleurinstelling van het deelvenster Gereedschappen of van een geselecteerde vorm.
Object Filter	Het object Filter bevat alle eigenschappen voor alle filters.
Object flash (fl)	Het object flash vertegenwoordigt de toepassing Flash.
Object FLfile	Met het object FLfile kunt u Flash-uitbreidingen schrijven die bestanden en mappen op het lokale systeem kunnen openen, wijzigen en verwijderen.
Object folderItem	Het object folderItem is een subklasse van het object Item.

Object	Beschrijving
Object fontItem	Het object fontItem is een subklasse van het object Item.
Object Frame	Het object Frame vertegenwoordigt frames in de laag.
Object HalfEdge	Gerichte zijde van de rand van een object Shape.
Object Instance	Het object Instance is een subklasse van het object Element.
Object Item	Het object Item is een abstracte basisklasse.
Object Layer	Het object Layer vertegenwoordigt een laag in de tijdlijn.
Object library	Het object library staat voor het deelvenster Bibliotheek.
Object Math	Het object Math is beschikbaar als alleen-lezen eigenschap van het object flash ( <code>fl.Math</code> ).
Object Matrix	Het object Matrix vertegenwoordigt een transformatiematrix.
Object outputPanel	Het object outputPanel vertegenwoordigt het deelvenster Uitvoer waarin informatie over probleemoplossing wordt weergegeven, bijvoorbeeld syntaxisfouten. Het is een eigenschap van het object flash ( <code>fl.outputPanel</code> ).
Object Oval	Het object Oval is een vorm die wordt getekend met het gereedschap Ovaal. Gebruik <code>shape.isOvalObject</code> om te bepalen of een item een object Oval is.
Object Parameter	Het objecttype Parameter wordt benaderd vanuit de array <code>screen.parameters</code> (die overeenkomt met het scherm Eigenschapcontrole in het ontwerp gereedschap van Flash) of door de array <code>componentInstance.parameters</code> (die overeenkomt met de component Eigenschapcontrole in het ontwerp gereedschap).
Object Path	Het object Path definieert een reeks lijnsegmenten (recht, gekromd of beide) die u in het algemeen gebruikt wanneer u uitbreidbare gereedschappen maakt.
Object presetItem	Het object presetItem vertegenwoordigt een item (voorstelling of map) in het deelvenster Voorinstellingen voor beweging.
Object presetPanel	Het object presetPanel vertegenwoordigt het deelvenster Voorinstellingen voor beweging (Venster > Voorinstellingen voor beweging). Het is een eigenschap van het object flash ( <code>fl.presetPanel</code> ).
Object Rectangle	Het object Rectangle is een vorm die wordt getekend met het gereedschap Rechthoek. Gebruik <code>shape.isRectangleObject</code> om te bepalen of een item een object Rectangle is.
Object Screen	Het object Screen vertegenwoordigt één scherm van een dia- of formulierdocument.
Object ScreenOutline	Het object ScreenOutline vertegenwoordigt de groep schermen in een dia- of formulierdocument.
Object Shape	Het object Shape is een subklasse van het object Element. Het object Shape biedt meer nauwkeurigheid dan de teken-API's voor het bewerken of maken van geometrische vormen in het werkgebied.
Object SoundItem	Het object SoundItem is een subklasse van het object Item. Het staat voor een bibliotheekitem dat wordt gebruikt om een geluid te maken.
Object Stroke	Het object Stroke bevat alle instellingen voor een streek, waaronder de aangepaste instellingen.
Object swfPanel	Het object swfPanel vertegenwoordigt een Window SWF-deelvenster. Window SWF-deelvensters zijn SWF-bestanden die toepassingen implementeren die u vanuit de ontwerpomgeving van Flash kunt uitvoeren. De array <code>swfPanel</code> -objecten is een eigenschap van het object flash ( <code>fl.swfPanels</code> ).
Object SymbolInstance	Het object SymbolInstance is een subklasse van het object Instance en vertegenwoordigt een symbool in een frame.

Object	Beschrijving
Object SymbolItem	Het object SymbolItem is een subklasse van het object Item.
Object Text	Het object Text vertegenwoordigt één tekstitem in een document.
Object TextAttrs	Het object TextAttrs bevat alle teksteigenschappen die op een subselectie kunnen worden toegepast. Dit object is een subklasse van het object Text.
Object TextRun	Het object TextRun vertegenwoordigt een groep opeenvolgende tekens met kenmerken die overeenkomen met alle eigenschappen in het object TextAttrs.
Object Timeline	Het object Timeline vertegenwoordigt de Flash-tijdlijn, die voor het huidige document toegankelijk is met <code>fl.getDocumentDOM().getTimeline()</code> .
Object ToolObj	Een object ToolObj vertegenwoordigt een individueel gereedschap in het deelvenster Gereedschappen.
Object Tools	Het object Tools kan worden gebruikt vanuit het object Flash ( <code>fl.tools</code> ).
Object Vertex	Het object Vertex is het deel van de gegevensstructuur van een vorm dat de coördinaatgegevens bevat.
Object VideoItem	Het object VideoItem is een subklasse van het object Item.
Object XMLUI	Met het object XMLUI kunt u eigenschappen van een XMLUI-dialoogvenster ophalen en instellen en een XMLUI-dialoogvenster accepteren of annuleren.

## Het Document Object Model van Flash

Het Document Object Model (DOM) voor de Flash JavaScript API bestaat uit een verzameling functies op hoofdniveau (zie “[Functies en methoden op hoofdniveau](#)” op pagina 15) en twee objecten op hoofdniveau: het object FLfile en het flash-object (fl). Elk object is gegarandeerd beschikbaar voor een script omdat het altijd bestaat wanneer de ontwerpomgeving van Flash geopend is. Zie [Object FLfile](#) en [Object flash \(fl\)](#) voor meer informatie.

Wanneer u naar het flash-object verwijst, kunt u `flash` of `fl` gebruiken. Als u bijvoorbeeld alle geopende FLA-bestanden wilt sluiten, kunt u een van de volgende instructies gebruiken:

```
flash.closeAll();
fl.closeAll();
```

Het flash-object bevat de volgende *onderliggende objecten*:

Object	Toegang verkrijgen
Object actionsPanel	Gebruik <code>fl.actionsPanel</code> om toegang te krijgen tot het object actionsPanel. Dit object komt overeen met het deelvenster Handelingen in de ontwerpomgeving van Flash.
Object compilerErrors	Gebruik <code>fl.compilerErrors</code> om toegang te krijgen tot het object compilerErrors. Dit object komt overeen met het deelvenster Compilerfouten in de ontwerpomgeving van Flash.
Object componentsPanel	Gebruik <code>fl.componentsPanel</code> om toegang te krijgen tot het object componentsPanel. Dit object komt overeen met het deelvenster Componenten in de ontwerpomgeving van Flash.
Object Document	Gebruik <code>fl.documents</code> om een array met alle geopende documenten op te halen; gebruik <code>fl.documents[index]</code> om toegang tot een specifiek document te krijgen; gebruik <code>fl.getDocumentDOM()</code> om toegang tot het huidige document te krijgen (het document met focus).
Object drawingLayer	Gebruik <code>fl.drawingLayer</code> om toegang te krijgen tot het object drawingLayer.
Object Math	Gebruik <code>fl.Math</code> om toegang te krijgen tot het object Math.
Object outputPanel	Gebruik <code>fl.outputPanel</code> om toegang te krijgen tot het object outputPanel. Dit object komt overeen met het deelvenster Uitvoer in de ontwerpomgeving van Flash.

Object	Toegang verkrijgen
Object <code>presetPanel</code>	Gebruik <code>fl.presetPanel</code> om toegang te krijgen tot het object <code>presetPanel</code> . Dit object komt overeen met het deelvenster Voorinstellingen voor beweging (Venster > Voorinstellingen voor beweging).
Object <code>swfPanel</code>	Gebruik <code>fl.swfPanels</code> om toegang te krijgen tot een array met <code>swfPanel</code> -objecten. Deze objecten komen overeen met Window SWF-deelvensters.
Object <code>Tools</code>	Gebruik <code>fl.tools</code> om toegang te krijgen tot een array met gereedschapobjecten.
Object XMLUI	Gebruik <code>fl.xmlui</code> om toegang te krijgen tot een object XML User Interface (XMLUI). Met het object XMLUI kunt u eigenschappen van een XMLUI-dialoogvenster ophalen en instellen.

## Het object Document

Een belangrijke eigenschap van het flash-object op hoofd niveau is de eigenschap `fl.documents`. Deze eigenschap bevat een array van Document-objecten, die allemaal staan voor een van de FLA-bestanden die op dit moment geopend zijn in de ontwerpomgeving. De eigenschappen van elk object Document staan voor de meeste elementen die een FLA-bestand kan bevatten. Daarom is een groot deel van de DOM samengesteld uit onderliggende objecten en eigenschappen van het object Document. Zie [Object Document](#) voor meer informatie.

Als u bijvoorbeeld naar het eerste geopende document wilt verwijzen, gebruikt u de instructie `flash.documents[0]` of `fl.documents[0]`. Het eerste document is het eerste Flash-document dat tijdens de huidige sessie is geopend in de ontwerpomgeving. Wanneer het eerst geopende document wordt gesloten, worden de indexen van de andere geopende documenten verlaagd.

Als u de index van een specifiek document zoekt, gebruikt u `flash.findDocumentIndex(nameOfDocument)` of `fl.findDocumentIndex(nameOfDocument)`. Zie [fl.findDocumentIndex\(\)](#).

Als u toegang wilt krijgen tot het document dat de focus heeft, gebruikt u de instructie `flash.getDocumentDOM()` of `fl.getDocumentDOM()`. Zie [fl.getDocumentDOM\(\)](#). De syntaxis van de laatste instructie wordt in de meeste voorbeelden in dit document gebruikt.

Als u een specifiek document wilt zoeken in de array `fl.documents`, doorloopt u de array en test u elk document op de eigenschap `document.name`. Zie [fl.documents](#) en [document.name](#).

Alle objecten in de DOM die niet in de vorige tabel staan (zie “[Het Document Object Model van Flash](#)” op pagina 9), zijn toegankelijk via het object Document. Als u bijvoorbeeld toegang wilt krijgen tot de bibliotheek van een document, gebruikt u de eigenschap `document.library` om een bibliotheekobject op te halen:

```
fl.getDocumentDOM().library
```

Als u toegang wilt verkrijgen tot de array van items in de bibliotheek, gebruikt u de eigenschap `library.items`. Elk element in de array is een object Item:

```
fl.getDocumentDOM().library.items
```

Als u toegang wilt krijgen tot een specifiek item in de bibliotheek, geeft u een lid van de array `library.items` op:

```
fl.getDocumentDOM().library.items[0]
```

Met andere woorden, het object bibliotheek is een onderliggend item van het object Document, en het object Item is een onderliggend item van het object bibliotheek. Zie [document.library](#), [Object library](#), [library.items](#) en [Object Item](#) voor meer informatie.

## Het doel van een handeling opgeven

Tenzij anders aangegeven, hebben methoden een effect op de huidige focus of selectie. Met het volgende script wordt bijvoorbeeld het formaat van de huidige selectie verdubbeld omdat er geen specifiek object is opgegeven:



**Inleiding**

```
fl.getDocumentDOM().scaleSelection(2, 2);
```

In sommige gevallen wilt u een handeling die zicht exact richt op het nu geselecteerde item in het Flash-document. Gebruik dan de array die de eigenschap `document.selection` bevat (zie [document.selection](#)). Het eerste element in de array staat voor het geselecteerde item, zoals in het volgende voorbeeld wordt getoond:

```
var accDescription = fl.getDocumentDOM().selection[0].description;
```

Met het volgende script wordt het formaat van het eerste element in het werkgebied verdubbeld dat is opgeslagen in de elementenarray, in plaats van de huidige selectie:

```
var element = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
if (element) {
    element.width = element.width*2;
    element.height = element.height*2;
}
```

U kunt bijvoorbeeld ook alle elementen op het werkgebied doorlopen en de breedte en hoogte met een bepaalde factor vergroten, zoals in het volgende voorbeeld wordt getoond:

```
var elementArray =
    fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
for (var i=0; i < elementArray.length; i++) {
    var offset = 10;
    elementArray[i].width += offset;
    elementArray[i].height += offset;
}
```

## Overzicht van de DOM-structuur

In de volgende lijst ziet u de DOM-structuur in hoofdpunten. Getallen aan het begin van elke regel geven het niveau van een object aan. Een object dat begint met 03 is een onderliggend element van het hoger gelegen object met 02, dat op zijn beurt weer een onderliggend element is van het hoger gelegen object dat begint met 01.

In sommige gevallen is een object beschikbaar als u een eigenschap van het bovenliggende object opgeeft. De eigenschap `document.timelines` bevat bijvoorbeeld een array met Timeline-objecten. Deze eigenschappen worden vermeld in het volgende overzicht.

Sommige objecten zijn subclasses van andere objecten in plaats van onderliggende elementen van andere objecten. Een object dat een subklasse van een ander object is, heeft zijn eigen methoden en/of eigenschappen, naast de methoden en eigenschappen van het bovenliggende object (de superklasse). Subklassen hebben hetzelfde niveau in de hiërarchie als hun superklasse. Het object `Item` is bijvoorbeeld een superklasse van het object `BitmapItem`. Deze relaties worden geïllustreerd in het volgende voorbeeld:

**Inleiding**

- 01 Top-Level Functions and Methods
- 01 FLfile object
- 01 flash object (fl)
  - 02 compilerErrors object
  - 02 componentsPanel object
  - 02 Document object (fl.documents array)
    - 03 Filter object
    - 03 Matrix object
    - 03 Fill object
    - 03 Stroke object
    - 03 library object
      - 04 Item object (library.items array)
      - 04 BitmapItem object (subclass of Item object)
      - 04 folderItem object (subclass of Item object)
      - 04 fontItem object (subclass of Item object)
      - 04 SoundItem object (subclass of Item object)
      - 04 SymbolItem object (subclass of Item object)
      - 04 VideoItem object (subclass of Item object)
  - 03 Timeline object (document.timelines array)
    - 04 Layer object (timeline.layers array)
      - 05 Frame object (layer.frames array)
        - 06 Element object (frame.elements array)
          - 07 Matrix object (element.matrix)
        - 06 Instance object (abstract class, subclass of Element object)
        - 06 BitmapInstance object (subclass of Instance object)
        - 06 CompiledClipInstance object (subclass of Instance object)
        - 06 ComponentInstance object (subclass of SymbolInstance object)
          - 07 Parameter object (componentInstance.parameters array)
        - 06 SymbolInstance object (subclass of Instance object)
        - 06 Text object (subclass of Element object)
          - 07 TextRun object (text.textRuns array)
            - 08 TextAttrs object (textRun.textAttrs array)
      - 06 Shape object (subclass of Element object)
        - 07 Oval object
        - 07 Rectangle object
        - 07 Contour object (shape.contours array)
          - 08 HalfEdge object
            - 09 Vertex object
            - 09 Edge object

```

    07 Edge object (shape.edges array)
      08 HalfEdge object
        09 Vertex object
          09 Edge object
    07 Vertex object (shape.vertices array)
      08 HalfEdge object
        09 Vertex object
          09 Edge object
    05 Parameter object (screen.parameters array)
02 drawingLayer object
  03 Path object
    04 Contour object
02 Math object
02 outputPanel object
02 presetPanel object
  03 presetItem object (presetPanel.items array)
02 swfPanel object
02 Tools object (fl.tools array)
  03 ToolObj object (tools.toolObjs array)
02 XMLUI object

```

## Voorbeeldimplementaties

Er zijn verschillende JSFL-voorbeeldimplementaties beschikbaar voor Adobe Flash Professional CS5 en CS5.5. U kunt deze bestanden bekijken en installeren om de JavaScript-API beter te leren kennen. De voorbeelden staan in een map met de naam Samples/ExtendingFlash in het bestand Samples.zip op [www.adobe.com/go/learn\\_fl\\_samples\\_nl](http://www.adobe.com/go/learn_fl_samples_nl).

### Voorbeeld van opdracht Shape

Een voorbeeldscript van de JavaScript-API met de naam Shape.jsfl vindt u in de map ExtendingFlash/Shape (zie "Voorbeeldimplementaties" hierboven). Met dit script wordt informatie over de contouren van de vorm in het deelvenster Uitvoer weergegeven.

#### Het script Shape installeren en uitvoeren:

- 1 Kopieer het bestand Shape.jsfl naar de map Configuration/Commands (zie "JSFL-bestanden opslaan" op pagina 2).
- 2 In een Flash-document (FLA-bestand) selecteert u een vormobject.
- 3 Selecteer Opdrachten > Vorm om het script uit te voeren.

### Voorbeeld van opdracht voor ophalen en instellen van filters

Een voorbeeldscript van de JavaScript-API met de naam filtersGetSet.jsfl vindt u in de map ExtendingFlash/filtersGetSet (zie "Voorbeeldimplementaties" hierboven). Met dit script worden filters aan een geselecteerd object toegevoegd en informatie over de filters weergegeven die in het deelvenster Uitvoer worden toegevoegd.

#### Het script filtersGetSet installeren en uitvoeren:

- 1 Kopieer het bestand filtersGetSet.jsfl naar de map Configuration/Commands (zie "JSFL-bestanden opslaan" op pagina 2).
- 2 In een Flash-document (FLA-bestand) selecteert u een tekst, filmclip of knopobject.

- 3 Selecteer Opdrachten > filtersGetSet om het script uit te voeren.

## Voorbeeld van gereedschap Veelhoek-Ster

Een voorbeeldscript van de JavaScript-API met de naam PolyStar.jsfl vindt u in de map ExtendingFlash/PolyStar (zie "Voorbeeldimplementaties" hierboven).

PolyStar.jsfl is een kopie van het gereedschap VeelhoekSter in het Flash-deelvenster Gereedschappen. Het script laat zien hoe u het gereedschap VeelhoekSter bouwt met de JavaScript-API en bevat gedetailleerde beschrijvingen van wat de code doet. Lees dit bestand om meer inzicht te krijgen in de manier waarop u de JavaScript-API kunt gebruiken. Lees ook het bestand PolyStar.xml in de map Tools voor meer informatie over het zelf bouwen van een gereedschap.

## Voorbeeld van deelvenster Bitmap overtrekken

Een set bestanden met de naam TraceBitmap fla en TraceBitmap.swf bevindt zich in de map ExtendingFlash/TraceBitmapPanel (zie "Voorbeeldimplementaties" hierboven). Deze bestanden laten zien hoe u een deelvenster ontwerpt en bouwt om de functies van Flash te beheren. U ziet ook het gebruik van de functie `MMExecute()` waarmee u JavaScript-opdrachten vanuit een ActionScript-script aanroept.

### Het voorbeeld TraceBitmap uitvoeren:

- 1 Als Flash geopend is, sluit u eerst Flash af.
- 2 Kopieer het bestand TraceBitmap.swf naar de map WindowSWF, een submap van de map Configuration (zie "JSFL-bestanden opslaan" op pagina 2). In Windows XP vindt u de map bijvoorbeeld in *opstartschijf:\Documents and Settings\gebruiker\Local Settings\Application Data\Adobe\Flash CS5\taal\Configuration\WindowSWF*.
- 3 Start Flash.
- 4 Maak of open een Flash-document (FLA-bestand) en importeer een bitmap- of JPEG-afbeelding in het bestand. U kunt het bestand flower.jpg in de map TraceBitmapPanel of een andere afbeelding gebruiken.
- 5 Selecteer de geïmporteerde afbeelding en selecteer Venster > Andere deelvensters > TraceBitmap.
- 6 Klik op Verzenden.  
De afbeelding wordt omgezet in een groep vormen.

## Voorbeeld-DLL

U vindt een voorbeeld van een DLL-implementatie in de map ExtendingFlash/dllSampleComputeSum (zie "Voorbeeldimplementaties" hierboven). Zie "Uitbreidbaarheid in C" op pagina 540 voor meer informatie over het bouwen van DLL's.

# Hoofdstuk 2: Functies en methoden op hoofdniveau

## Over deze sectie

In deze sectie worden de functies en methoden op hoofdniveau beschreven die beschikbaar zijn wanneer u de Adobe Flash JavaScript-API (programmeerinterface voor toepassingen) gebruikt. Zie “[JSFL-bestanden opslaan](#)” op pagina 2 voor meer informatie over waar u JavaScript API-bestanden moet opslaan.

## Algemene methoden

De volgende methoden kunnen worden aangeroepen vanuit elk JavaScript API-script:

```
alert ()
confirm ()
prompt ()
```

## Uitbreidbare gereedschappen

De volgende functies zijn beschikbaar in scripts die uitbreidbare gereedschappen maken:

```
activate ()
configureTool ()
deactivate ()
keyDown ()
keyUp ()
mouseDoubleClick ()
mouseDown ()
mouseMove ()
mouseUp ()
notifySettingsChanged ()
setCursor ()
```

## activate()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function activate () {
    // statements
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief wordt (als het gereedschap wordt geselecteerd in het deelvenster Gereedschappen). Gebruik deze functie om initialisatietaken uit te voeren die vereist zijn voor het gereedschap.

### Voorbeeld

In het volgende voorbeeld wordt de waarde `tools.activeTool` ingesteld wanneer het uitbreidbare gereedschap wordt geselecteerd in het deelvenster Gereedschappen:

```
function activate() {  
    var theTool = fl.tools.activeTool  
}
```

### Zie ook

`tools.activeTool`

## alert()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
alert ( alertText )
```

### Parameters

**alertText** Een tekenreeks die het bericht opgeeft dat u wilt weergeven in het dialoogvenster Waarschuwing.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een tekenreeks weergegeven in een modaal dialoogvenster Waarschuwing, samen met de knop OK.

### Voorbeeld

In het volgende voorbeeld wordt het bericht "Process Complete" weergegeven in een dialoogvenster Waarschuwing:

```
alert("Process Complete");
```

### Zie ook

`confirm()`, `prompt()`

## configureTool()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function configureTool() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer Flash wordt geopend en het uitbreidbare gereedschap wordt geladen in het deelvenster Gereedschappen. Gebruik deze functie om de informatie in te stellen die Flash nodig heeft voor dit gereedschap.

### Voorbeeld

In het volgende voorbeeld worden twee mogelijke implementaties getoond van deze functie:

```
function configureTool() {  
    theTool = fl.tools.activeTool;  
    theTool.setToolName("myTool");  
    theTool.setIcon("myTool.png");  
    theTool.setMenuString("My Tool's menu string");  
    theTool.setToolTip("my tool's tool tip");  
    theTool.setOptionsFile( "mtTool.xml" );  
}
```

```
function configureTool() {  
    theTool = fl.tools.activeTool;  
    theTool.setToolName("ellipse");  
    theTool.setIcon("Ellipse.png");  
    theTool.setMenuString("Ellipse");  
    theTool.setToolTip("Ellipse");  
    theTool.showTransformHandles( true );  
}
```

## confirm()

### Beschikbaarheid

Flash 8.

### Gebruik

```
confirm ( strAlert )
```

### Parameters

**strAlert** Een tekenreeks die het bericht opgeeft dat u wilt weergeven in het dialoogvenster Waarschuwing.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de gebruiker op OK klikt; `false` als de gebruiker op Annuleren klikt.

### Beschrijving

Methode; hiermee wordt een tekenreeks weergegeven in een modaal dialoogvenster Waarschuwing, samen met de knoppen OK en Annuleren.

*Opmerking: Als er geen documenten (FLA-bestanden) geopend zijn, mislukt deze methode met een fout.*

### Voorbeeld

In het volgende voorbeeld wordt het bericht "Sort data?" weergegeven in het dialoogvenster Waarschuwing:

```
confirm("Sort data?");
```

### Zie ook

[alert\(\)](#), [prompt\(\)](#)

## deactivate()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function deactivate() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap inactief wordt (terwijl dit gereedschap actief is, wordt een ander gereedschap geselecteerd). Gebruik deze functie om opschoontaken uit te voeren die vereist zijn voor het gereedschap.

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven in het deelvenster Uitvoer wanneer het gereedschap inactief wordt:

```
function deactivate() {  
    fl.trace( "Tool is no longer active" );  
}
```



## keyDown()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function keyDown() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en de gebruiker op een toets drukt. Het script moet [tools.getKeyDown\(\)](#) aanroepen om te bepalen welke toets werd ingedrukt.

### Voorbeeld

In het volgende voorbeeld wordt informatie weergegeven over welke toets werd ingedrukt wanneer het uitbreidbare gereedschap actief is en de gebruiker op een toets drukt.

```
function keyDown() {  
    fl.trace("key " + fl.tools.getKeyDown() + " was pressed");  
}
```

### Zie ook

[keyUp\(\)](#), [tools.getKeyDown\(\)](#)

## keyUp()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function keyUp() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en een toets wordt losgelaten.

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven in het deelvenster Uitvoer wanneer het uitbreidbare gereedschap actief is en een toets wordt losgelaten:

```
function keyUp() {  
    fl.trace("Key is released");  
}
```

### Zie ook

[keyDown\(\)](#)

## mouseDoubleClick()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function mouseDoubleClick() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en met de muis dubbelgeklikt wordt op het werkgebied.

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven in het deelvenster Uitvoer wanneer het uitbreidbare gereedschap actief is en dubbelgeklikt wordt met de muis.

```
function mouseDoubleClick() {  
    fl.trace("Mouse was double-clicked");  
}
```

## mouseDown()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function mouseDown( [ pt ] ) {  
    // statements  
}
```

### Parameters

**pt** Een punt dat de locatie van de muis opgeeft wanneer met de muis wordt geklikt. De locatie wordt doorgegeven aan de functie wanneer met de muis wordt geklikt. Deze parameter is optioneel.

### Geretoureerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en met de muis geklikt wordt terwijl de aanwijzer zich boven het werkgebied bevindt.

### Voorbeeld

In de volgende voorbeelden wordt getoond hoe deze functie kan worden gebruikt als het uitbreidbare gereedschap actief is. In het eerste voorbeeld wordt in het deelvenster Uitvoer het bericht weergegeven dat de muisknop werd ingedrukt. In het tweede voorbeeld worden de *x*- en *y*-coördinaat weergegeven van de locatie van de muis op het moment dat werd geklikt.

```
function mouseDown() {  
    fl.trace("Mouse button has been pressed");  
}  
function mouseDown(pt) {  
    fl.trace("x = "+ pt.x+" :: y = "+pt.y);  
}
```

## mouseMove()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function mouseMove( [ pt ] ) {  
    // statements  
}
```

### Parameters

**pt** Een tekenreeks die de huidige locatie van de muis opgeeft. De locatie wordt doorgegeven aan de functie wanneer de muis beweegt. Zo wordt de locatie van de muis gevolgd. Als het werkgebied wordt weergegeven in de modus Bewerken of Op plaats bewerken, zijn de puntcoördinaten ten opzichte van het object dat wordt bewerkt. Anders zijn de puntcoördinaten ten opzichte van het werkgebied. Deze parameter is optioneel.

### Geretoureerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en de muis wordt bewogen over een opgegeven punt in het werkgebied. De muisknop kan wel of niet zijn ingedrukt.

### Voorbeeld

In de volgende voorbeelden wordt getoond hoe deze functie kan worden gebruikt. In het eerste voorbeeld wordt in het deelvenster Uitvoer het bericht weergegeven dat de muis wordt verplaatst. In het tweede voorbeeld worden de  $x$ - en  $y$ -coördinaat weergegeven van de locatie van de muis terwijl deze beweegt.

```
function mouseMove() {
    fl.trace("moving");
}

function mouseMove(pt) {
    fl.trace("x = " + pt.x + " :: y = " + pt.y);
}
```

## mouseUp()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function mouseUp() {
    // statements
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en de muisknop wordt losgelaten nadat is geklikt op het werkgebied.

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven in het deelvenster Uitvoer wanneer het uitbreidbare gereedschap actief is en de muisknop wordt losgelaten.

```
function mouseUp() {
    fl.trace("mouse is up");
}
```

## notifySettingsChanged()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function notifySettingsChanged() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en de gebruiker de opties wijzigt in de Eigenschapcontrole. U kunt de eigenschap `tools.activeTool` gebruiken om de huidige waarden van de opties op te vragen (zie [tools.activeTool](#)).

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven in het deelvenster Uitvoer wanneer het uitbreidbare gereedschap actief is en de gebruiker de opties wijzigt in de Eigenschapcontrole.

```
function notifySettingsChanged() {  
    var theTool = fl.tools.activeTool;  
    var newValue = theTool.myProp;  
}
```

## prompt()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
prompt(promptMsg [,text])
```

### Parameters

**promptMsg** Een tekenreeks die wordt weergegeven in het dialoogvenster Vragen (maximaal 256 tekens in Mac OS X).

**text** Een optionele tekenreeks die wordt weergegeven als standaardwaarde voor het tekstveld.

### Geretourneerde waarde

De tekenreeks die de gebruiker heeft getypt als de gebruiker op OK klikt; `null` als de gebruiker op Annuleren klikt.

### Beschrijving

Methode; hiermee worden een vraag en een optionele tekst weergegeven in een modaal dialoogvenster Waarschuwing, samen met de knoppen OK en Annuleren.

### Voorbeeld

In het volgende voorbeeld wordt de gebruiker gevraagd een gebruikersnaam in te voeren. Als de gebruiker een naam typt en op OK klikt, verschijnt de naam in het deelvenster Uitvoer.

```
var userName = prompt("Enter user name", "Type user name here");  
fl.trace(userName);
```

### Zie ook

[alert\(\)](#), [confirm\(\)](#)

## setCursor()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
function setCursor() {  
    // statements  
}
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Functie; wordt aangeroepen wanneer het uitbreidbare gereedschap actief is en de muis wordt bewogen om het script in staat te stellen aangepaste aanwijzers in te stellen. Het script moet `tools.setCursor()` aanroepen om te bepalen welke aanwijzer moet worden gebruikt. Zie [tools.setCursor\(\)](#) voor een lijst waarin wordt aangegeven welke aanwijzers overeenkomen met welke gehele getallen.

### Voorbeeld

```
function setCursor() {  
    fl.tools.setCursor( 1 );  
}
```

# Hoofdstuk 3: Object actionsPanel

## Beschikbaarheid

Flash CS3 Professional.

## Beschrijving

Het object actionsPanel dat het momenteel weergegeven deelvenster Handelingen vertegenwoordigt, is een eigenschap van het object flash (zie [fl.actionsPanel](#)).

## Overzicht van methoden

U kunt de volgende methoden gebruiken met het object actionsPanel:

Methoden	Beschrijving
<code>actionsPanel.getClassForObject()</code>	Hiermee wordt de klasse van een opgegeven variabele geretourneerd.
<code>actionsPanel.getScriptAssistMode()</code>	Hiermee wordt aangegeven of de modus Scriptassistentie is ingeschakeld.
<code>actionsPanel.getSelectedText()</code>	Hiermee wordt de tekst geretourneerd die momenteel is geselecteerd in het deelvenster Handelingen.
<code>actionsPanel.getText()</code>	Hiermee wordt de tekst in het deelvenster Handelingen geretourneerd.
<code>actionsPanel.hasSelection()</code>	Hiermee wordt aangegeven of er momenteel tekst is geselecteerd in het deelvenster Handelingen.
<code>actionsPanel.replaceSelectedText()</code>	Hiermee wordt de momenteel geselecteerde tekst vervangen door opgegeven tekst.
<code>actionsPanel.setScriptAssistMode()</code>	Hiermee wordt de modus Scriptassistentie in- of uitgeschakeld.
<code>actionsPanel.setSelection()</code>	Hiermee wordt een opgegeven set tekens in het deelvenster Handelingen geselecteerd.
<code>actionsPanel.setText()</code>	Hiermee wordt alle tekst in het deelvenster Handelingen gewist en wordt vervolgens opgegeven tekst toegevoegd.

## actionsPanel.getClassForObject()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.getClassForObject(ASvariableName)
```

### Parameters

**ASvariableName** Een tekenreeks die de naam van een ActionScript-variabele vertegenwoordigt.

### Geretourneerde waarde

Een tekenreeks die de klasse vertegenwoordigt waarvan *ASvariableName* een lid is.

### Beschrijving

Methode; hiermee wordt de klasse geretourneerd van de opgegeven variabele. Deze moet worden gedefinieerd in het momenteel weergegeven deelvenster Handelingen. Bovendien moet de cursor of geselecteerde tekst in het deelvenster Handelingen na de definitie van de variabele worden geplaatst.

### Voorbeeld

In het volgende voorbeeld wordt de klasse weergegeven die werd toegewezen aan de variabele myVar als u de cursor na de instructie var myVar:ActivityEvent plaatst; in het deelvenster Handelingen.

```
// Place the following code in the Actions panel,  
// and position the cursor somewhere after the end of the line  
var myVar:ActivityEvent;  
// Place the following code in the JSFL file  
var theClass = fl.actionsPanel.getClassForObject("myVar");  
fl.trace(theClass); // traces: "ActivityEvent"
```

## actionsPanel.getScriptAssistMode()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.getScriptAssistMode()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde die aangeeft of de modus Scriptassistentie is in- (`true`) of uitgeschakeld (`false`).

### Beschrijving

Methode; hiermee wordt aangegeven of de modus Scriptassistentie is ingeschakeld.

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven als de modus Scriptassistentie niet is ingeschakeld.

```
mAssist = fl.actionsPanel.getScriptAssistMode();  
if (!mAssist) {  
    alert("For more guidance when writing ActionScript code, try Script Assist mode");  
}
```

### Zie ook

[actionsPanel.setScriptAssistMode\(\)](#)



## `actionsPanel.getSelectedText()`

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.getSelectedText()
```

### Parameters

Geen.

### Geretourneerde waarde

Een tekenreeks die de tekst bevat die momenteel is geselecteerd in het deelvenster Handelingen.

### Beschrijving

Methode; hiermee wordt de tekst geretourneerd die momenteel is geselecteerd in het deelvenster Handelingen.

### Voorbeeld

In het volgende voorbeeld wordt de tekst weergegeven die momenteel is geselecteerd in het deelvenster Handelingen.

```
var apText = fl.actionsPanel.getSelectedText();  
fl.trace(apText);
```

### Zie ook

[actionsPanel.getText\(\)](#), [actionsPanel.hasSelection\(\)](#), [actionsPanel.replaceSelectedText\(\)](#),  
[actionsPanel.setSelection\(\)](#)

## `actionsPanel.getText()`

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.getText()
```

### Parameters

Geen.

### Geretourneerde waarde

Een tekenreeks die alle tekst in het deelvenster Handelingen bevat.

### Beschrijving

Methode; hiermee wordt de tekst in het deelvenster Handelingen geretourneerd.

### Voorbeeld

In het volgende voorbeeld wordt de tekst weergegeven die in het deelvenster Handelingen staat.

```
var apText = fl.actionsPanel.getText();  
fl.trace(apText);
```

#### Zie ook

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.setText\(\)](#)

## actionsPanel.hasSelection()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.hasSelection()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde die aangeeft of er tekst is geselecteerd in het deelvenster Handelingen (`true`) of dat de selectie ervan ongedaan is gemaakt (`false`).

### Beschrijving

Methode; hiermee wordt aangegeven of er momenteel tekst is geselecteerd in het deelvenster Handelingen.

### Voorbeeld

In het volgende voorbeeld wordt de tekst weergegeven die momenteel is geselecteerd in het deelvenster Handelingen. Als er geen tekst is geselecteerd, wordt alle tekst in het deelvenster Handelingen weergegeven.

```
if (fl.actionsPanel.hasSelection()) {  
    var apText = fl.actionsPanel.getSelectedText();  
}  
else {  
    var apText = fl.actionsPanel.getText();  
}  
fl.trace(apText);
```

#### Zie ook

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.getText\(\)](#), [actionsPanel.replaceSelectedText\(\)](#), [actionsPanel.setSelection\(\)](#)

## actionsPanel.replaceSelectedText()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.replaceSelectedText(replacementText)
```

### Parameters

**replacementText** Een tekenreeks die tekst vertegenwoordigt die de geselecteerde tekst in het deelvenster Handelingen vervangt.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het deelvenster Handelingen is gevonden, anders `false`.

### Beschrijving

Methode; hiermee wordt de momenteel geselecteerde tekst vervangen door de in *replacementText* opgegeven tekst. Als *replacementText* meer tekens bevat dan de geselecteerde tekst, komen alle tekens na de geselecteerde tekst nu ook achter *replacementText* te staan; ze worden dus niet overschreven.

### Voorbeeld

In het volgende voorbeeld wordt de tekst vervangen die momenteel is geselecteerd in het deelvenster Handelingen.

```
if (fl.actionsPanel.hasSelection()) {  
    fl.actionsPanel.replaceSelectedText("// © 2006 Adobe Inc.");  
}
```

### Zie ook

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.hasSelection\(\)](#), [actionsPanel.setSelection\(\)](#),  
[actionsPanel.setText\(\)](#)

## actionsPanel.setScriptAssistMode()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.setScriptAssistMode(bScriptAssist)
```

### Parameters

**bScriptAssist** Een Booleaanse waarde die aangeeft of de modus Scriptassistentie moet worden in- of uitgeschakeld.

### Geretourneerde waarde

Een Booleaanse waarde die aangeeft of de modus Scriptassistentie is in- of uitgeschakeld.

### Beschrijving

Methode; hiermee wordt de modus Scriptassistentie in- of uitgeschakeld.

### Voorbeeld

In het volgende voorbeeld wordt de status van de modus Scriptassistentie geschakeld.

```
fl.trace(fl.actionsPanel.getScriptAssistMode());  
if (fl.actionsPanel.getScriptAssistMode()) {  
    fl.actionsPanel.setScriptAssistMode(false);  
}  
else {  
    fl.actionsPanel.setScriptAssistMode(true);  
}  
fl.trace(fl.actionsPanel.getScriptAssistMode());
```

### Zie ook

[actionsPanel.getScriptAssistMode\(\)](#)

## actionsPanel.setSelection()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.setSelection(startIndex, numberOfChars)
```

### Parameters

**startIndex** Een op nul gebaseerde index die het eerste teken aangeeft dat moet worden geselecteerd.

**numberOfChars** Een geheel getal dat aangeeft hoeveel tekens moeten worden geselecteerd.

### Geretoureerde waarde

Een Booleaanse waarde die aangeeft of de gevraagde tekens mogen worden geselecteerd (`true`) of dat de selectie ervan ongedaan moet worden gemaakt (`false`).

### Beschrijving

Methode; hiermee wordt een opgegeven set tekens in het deelvenster Handelingen geselecteerd.

### Voorbeeld

In het volgende voorbeeld worden de tekens “2006” in het deelvenster Handelingen vervangen door de opgegeven tekst.

```
// Type the following as the first line in the Actions panel  
// 2006 - Addresses user request 40196  
// Type the following in the JSFL file  
fl.actionsPanel.setSelection(3,4);  
fl.actionsPanel.replaceSelectedText("// Last updated: 2007");
```

### Zie ook

[actionsPanel.getSelectedText\(\)](#), [actionsPanel.hasSelection\(\)](#),  
[actionsPanel.replaceSelectedText\(\)](#)

## `actionsPanel.setText()`

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
actionsPanel.setText(replacementText)
```

### Parameters

**replacementText** Een tekenreeks die tekst vertegenwoordigt die in het deelvenster Handelingen moet worden geplaatst.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de opgegeven tekst in het deelvenster Handelingen is geplaatst, anders `false`.

### Beschrijving

Methode; hiermee wordt alle tekst in het deelvenster Handelingen gewist en wordt vervolgens de in *replacementText* opgegeven tekst toegevoegd.

### Voorbeeld

In het volgende wordt de tekst in het deelvenster Handelingen vervangen door de opgegeven tekst.

```
fl.actionsPanel.setText("// Deleted this code - no longer needed");
```

### Zie ook

[actionsPanel.getText\(\)](#), [actionsPanel.replaceSelectedText\(\)](#)

# Hoofdstuk 4: Object BitmapInstance

**Overerving** [Object Element](#) > [Object Instance](#) > Object BitmapInstance

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object BitmapInstance is een subklasse van het object Instance en vertegenwoordigt een bitmap in een frame (zie [Object Instance](#)).

## Overzicht van methoden

Naast de methoden van het [Object Instance](#) kunt u de volgende methoden gebruiken voor het object BitmapInstance:

Methode	Beschrijving
<code>bitmapInstance.getBits()</code>	Hiermee kunt u bitmapeffecten maken door de bits uit de bitmap op te halen, deze te bewerken en vervolgens te retourneren aan Flash.
<code>bitmapInstance.setBits()</code>	Hiermee worden de bits van een bestaand bitmapelement ingesteld.

## Overzicht van eigenschappen

Naast de eigenschappen van het [Object Instance](#) kunt u de volgende eigenschappen gebruiken voor het object BitmapInstance:

Eigenschap	Beschrijving
<code>bitmapInstance.hPixels</code>	Alleen-lezen; een geheel getal dat de breedte van de bitmap in pixels vertegenwoordigt.
<code>bitmapInstance.vPixels</code>	Alleen-lezen; een geheel getal dat de hoogte van de bitmap in pixels vertegenwoordigt.

## bitmapInstance.getBits()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
bitmapInstance.getBits()
```

### Parameters

Geen.

### Geretourneerde waarde

Een object dat de elementen `width`, `height`, `depth` en `bits` vertegenwoordigt. Als de bitmap een kleurentabel heeft, vertegenwoordigt dit object ook de `cTab`-eigenschappen. Het element `bits` is een array met bytes. Het element `cTab` is een array met kleurwaarden in de notatie "#RRGGBB". De lengte van de array is de lengte van de kleurentabel.

De byte-array is alleen relevant als ernaar wordt verwezen door een DLL-bestand of gedeelde bibliotheek. Normaalgesproken gebruikt u deze alleen voor het maken van een uitbreidbaar gereedschap of effect. Zie "Uitbreidbaarheid in C" op pagina 540 voor meer informatie over het maken van DLL-bestanden voor gebruik met Flash JavaScript

### Beschrijving

Methode; hiermee kunt u bitmapeffecten maken door de bits uit de bitmap op te halen, deze te bewerken en vervolgens te retourneren aan Flash.

### Voorbeeld

Met de volgende code wordt een verwijzing gemaakt naar het momenteel geselecteerde object; getest of het object een bitmap is; en worden de hoogte, breedte en bitdiepte van de bitmap getraceerd:

```
var isBitmap = fl.getDocumentDOM().selection[0].instanceType;
if (isBitmap == "bitmap") {
    var bits = fl.getDocumentDOM().selection[0].getBits();
    fl.trace("height = " + bits.height);
    fl.trace("width = " + bits.width);
    fl.trace("depth = " + bits.depth);
}
```

### Zie ook

[bitmapInstance.setBits\(\)](#)

## bitmapInstance.hPixels

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`bitmapInstance.hPixels`

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat de breedte van de bitmap vertegenwoordigt. Dit is het aantal pixels in de horizontale afmeting.

### Voorbeeld

Met de volgende code wordt de breedte van de bitmap in pixels opgehaald:

```
// Get the number of pixels in the horizontal dimension.
var bmObj = fl.getDocumentDOM().selection[0];
var isBitmap = bmObj.instanceType;
if (isBitmap == "bitmap") {
    var numHorizontalPixels = bmObj.hPixels;
}
```

### Zie ook

[bitmapInstance.vPixels](#)

## bitmapInstance.setBits()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
bitmapInstance.setBits(bitmap)
```

### Parameters

**bitmap** Een object dat de eigenschappen `height`, `width`, `depth` en `bits` en `cTab` bevat. De eigenschappen `height`, `width` en `depth` zijn gehele getallen. De eigenschap `bits` is een array met bytes. De eigenschap `cTab` is alleen vereist voor bitmaps met een bitdiepte van 8 of minder en is een tekenreeks die een kleurwaarde vertegenwoordigt in de notatie "#RRGGBB".

***Opmerking:** De byte-array is alleen relevant als ernaar wordt verwezen door een externe bibliotheek. Normaalgesproken gebruikt u deze alleen voor het maken van een uitbreidbaar gereedschap of effect.*

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee stelt u de bits van een bestaand bitmapelement in. Hiermee kunt u bitmapeffecten maken door de bits uit de bitmap op te halen, deze te bewerken en de bitmap vervolgens te retourneren aan Flash.

### Voorbeeld

Met de volgende code wordt getest of de huidige selectie een bitmap is en wordt vervolgens de hoogte van de bitmap ingesteld op 150 pixels:

```
var isBitmap = fl.getDocumentDOM().selection[0].instanceType;
if (isBitmap == "bitmap") {
    var bits = fl.getDocumentDOM().selection[0].getBits();
    bits.height = 150;
    fl.getDocumentDOM().selection[0].setBits(bits);
}
```

### Zie ook

[bitmapInstance.getBits\(\)](#)

## bitmapInstance.vPixels

### Beschikbaarheid

Flash MX 2004.



### Gebruik

`bitmapInstance.vPixels`

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat de hoogte van de bitmap vertegenwoordigt. Dit is het aantal pixels in de verticale afmeting.

### Voorbeeld

Met de volgende code wordt de hoogte van de bitmap in pixels opgehaald.

```
// Get the number of pixels in the vertical dimension.  
var bmObj = fl.getDocumentDOM().selection[0];  
var isBitmap = bmObj.instanceType;  
if (isBitmap == "bitmap"){  
    var numVerticalPixels = bmObj.vPixels;  
}
```

### Zie ook

[bitmapInstance.hPixels](#)

# Hoofdstuk 5: Object BitmapItem

**Overerving** [Object Item](#) > object BitmapItem

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Een object BitmapItem verwijst naar een bitmap in de bibliotheek van een document. Het object BitmapItem is een subklasse van het object Item (zie [Object Item](#)).

## Overzicht van eigenschappen

Het object BitmapItem heeft naast de eigenschappen van het [Object Item](#) ook de volgende eigenschappen:

Eigenschap	Beschrijving
<code>bitmapItem.allowSmoothing</code>	Een Booleaanse waarde die aangeeft of een bitmap vloeiend gemaakt mag worden.
<code>bitmapItem.compressionType</code>	Een tekenreeks die het type afbeeldingscompressie bepaalt dat wordt toegepast op de bitmap.
<code>bitmapItem.fileLastModifiedDate</code>	Het aantal seconden dat tussen 1 januari 1970 en de wijzigingsdatum van het oorspronkelijke bestand is verstreken.
<code>bitmapItem.originalCompressionType</code>	Geeft aan of het item als een JPEG-bestand is geïmporteerd.
<code>bitmapItem.sourceFileExists</code>	Geeft aan of het in de bibliotheek geïmporteerde bestand nog bestaat op de locatie vanwaaruit het is geïmporteerd.
<code>bitmapItem.sourceFileIsCurrent</code>	Geeft aan of de bestandswijzigingsdatum van het bibliotheekitem dezelfde is als de wijzigingsdatum op de schijf van het geïmporteerde bestand.
<code>bitmapItem.sourceFilePath</code>	Het pad en de naam van het bestand dat in de bibliotheek is geïmporteerd.
<code>bitmapItem.useDeblocking</code>	Geeft aan of deblokkering is ingeschakeld.
<code>bitmapItem.useImportedJPEGQuality</code>	Een Booleaanse waarde die aangeeft of de standaardkwaliteit van JPEG voor importeren moet worden gebruikt.

## Overzicht van methoden

Het object BitmapItem heeft naast de eigenschappen van het [Object Item](#) ook de volgende methoden:

Methode	Beschrijving
<code>bitmapItem.exportToFile()</code>	Exporteert het opgegeven item naar een PNG- of JPG-bestand.

## bitmapItem.allowSmoothing

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
bitmapItem.allowSmoothing
```

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of het is toegestaan een bitmap vloeiender te maken (`true`) of niet (`false`).

### Voorbeeld

Met de volgende code wordt de eigenschap `allowSmoothing` van het eerste item in de bibliotheek van het huidige document ingesteld op `true`:

```
fl.getDocumentDOM().library.items[0].allowSmoothing = true;  
alert(fl.getDocumentDOM().library.items[0].allowSmoothing);
```

## bitmapItem.compressionType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
bitmapItem.compressionType
```

### Beschrijving

Eigenschap; een tekenreeks die het type afbeeldingscompressie bepaalt dat wordt toegepast op de bitmap. Acceptabele waarden zijn "photo" en "lossless". Als de waarde van `bitmapItem.useImportedJPEGQualityfalse` is, komt "photo" overeen met JPEG met een kwaliteit van 0 tot en met 100; als `bitmapItem.useImportedJPEGQualitytrue` is, komt "photo" overeen met JPEG waarvoor de standaarddocumentkwaliteit wordt gebruikt. De waarde "lossless" komt overeen met de indeling GIF of PNG (zie [bitmapItem.useImportedJPEGQuality](#)).

### Voorbeeld

Met de volgende code wordt de eigenschap `compressionType` van het eerste item in de bibliotheek van het huidige document ingesteld op waar "photo":

```
fl.getDocumentDOM().library.items[0].compressionType = "photo";  
alert(fl.getDocumentDOM().library.items[0].compressionType);
```

## bitmapItem.exportToFile()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.exportToFile(fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad en de naam van het geëxporteerde bestand aangeeft.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het bestand is geëxporteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven item geëxporteerd naar een PNG- of JPG-bestand.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een bitmapitem is, wordt dit item met de volgende code geëxporteerd als JPG-bestand:

```
var imageFileURL = "file:///C:/exportTest/out.jpg";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFile(imageFileURL);
```

## bitmapItem.fileLastModifiedDate

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.fileLastModifiedDate
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die een hexadecimaal getal bevat dat het aantal seconden vertegenwoordigt dat is verstreken tussen 1 januari 1970 en de wijzigingsdatum van het oorspronkelijke bestand op het moment dat het bestand werd geïmporteerd naar de bibliotheek. Als het bestand niet meer bestaat, is deze waarde "00000000".

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een bitmapitem is, wordt met de volgende code een hexadecimaal getal weergegeven zoals hierboven beschreven.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

### Zie ook

[bitmapItem.sourceFileExists](#), [bitmapItem.sourceFileIsCurrent](#), [bitmapItem.sourceFilePath](#), [FLfile.getModificationDate\(\)](#)

## bitmapItem.originalCompressionType

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.originalCompressionType
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die opgeeft of het opgegeven item is geïmporteerd als JPEG-bestand. Mogelijke waarden voor deze eigenschap zijn "photo" (voor JPEG-bestanden) en "lossless" (voor niet-gecomprimeerde bestandstypen zoals GIF en PNG).

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een bitmapitem is, wordt met de volgende code "photo" weergegeven als het bestand als JPEG-bestand naar de bibliotheek werd geïmporteerd, en "lossless" als dit niet het geval is:

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Imported compression type = "+ libItem.originalCompressionType);
```

### Zie ook

[bitmapItem.compressionType](#)

## bitmapItem.quality

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
bitmapItem.quality
```

### Beschrijving

Eigenschap; een geheel getal dat de kwaliteit van de bitmap opgeeft. Als u de standaarddocumentkwaliteit wilt gebruiken, geeft u -1 op; geef anders een geheel getal op van 0 tot en met 100. Alleen beschikbaar voor JPEG-compressie.

### Voorbeeld

Met de volgende code wordt de eigenschap `quality` van het eerste item in de bibliotheek van het huidige document ingesteld op 65:

```
fl.getDocumentDOM().library.items[0].quality = 65;  
alert(fl.getDocumentDOM().library.items[0].quality);
```

## bitmapItem.sourceFileExists

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.sourceFileExists
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde: `true` als het bestand dat naar de bibliotheek is geïmporteerd, nog bestaat op de locatie vanwaar het werd geïmporteerd, anders `false`.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een bitmapitem is, wordt met de volgende code "true" weergegeven als het bestand dat naar de bibliotheek werd geïmporteerd, nog bestaat.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

### Zie ook

[bitmapItem.sourceFileIsCurrent](#),

[bitmapItem.sourceFilePath](#)

## bitmapItem.sourceFileIsCurrent

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.sourceFileIsCurrent
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde: `true` als de wijzigingsdatum van het bibliotheekitem gelijk is aan de wijzigingsdatum op schijf van het bestand dat is geïmporteerd, anders `false`.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een bitmapitem is, wordt met de volgende code "true" weergegeven als het bestand dat werd geïmporteerd, niet op schijf is gewijzigd sinds het werd geïmporteerd:

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

### Zie ook

[bitmapItem.fileLastModifiedDate](#), [bitmapItem.sourceFilePath](#)

## bitmapItem.sourceFilePath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.sourceFilePath
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks, uitgedrukt als bestand:/// URI, die het pad en de naam vertegenwoordigt van het bestand dat naar de bibliotheek is geïmporteerd.

### Voorbeeld

In het onderstaande voorbeeld worden de naam en het pad naar het bronbestand weergegeven van alle items in de bibliotheek van het type "bitmap":

```
for (idx in fl.getDocumentDOM().library.items) {  
  if (fl.getDocumentDOM().library.items[idx].itemType == "bitmap") {  
    var myItem = fl.getDocumentDOM().library.items[idx];  
    fl.trace(myItem.name + " source is " + myItem.sourceFilePath);  
  }  
}
```

### Zie ook

[bitmapItem.sourceFileExists](#)

## bitmapItem.useDeblocking

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
bitmapItem.useDeblocking
```

### Beschrijving

Eigenschap: een Booleaanse waarde die opgeeft of deblokkering is ingeschakeld (`true`) of niet (`false`).

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een bitmapitem is, wordt met de volgende code deblokkering ingeschakeld voor het item:

```
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.useDeblocking = true;
```

## bitmapItem.useImportedJPEGQuality

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`bitmapItem.useImportedJPEGQuality`

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of de standaardkwaliteit van geïmporteerde JPEG-bestanden moet worden gebruikt (`true`) of niet (`false`). Alleen beschikbaar voor JPEG-compressie.

### Voorbeeld

Met de volgende code wordt de eigenschap `useImportedJPEGQuality` van het eerste item in de bibliotheek van het huidige document ingesteld op waar `true`:

```
fl.getDocumentDOM().library.items[0].useImportedJPEGQuality = true;  
alert(fl.getDocumentDOM().library.items[0].useImportedJPEGQuality);
```



# Hoofdstuk 6: Object CompiledClipInstance

**Overerving** [Object Element](#) > [Object Instance](#) > Object CompiledClipInstance

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object CompiledClipInstance is een subklasse van het object Instance. Het is in wezen een instantie van een filmclip die is geconverteerd naar een bibliotheekitem met een gecompileerde clip (zie [Object Instance](#)).

## Overzicht van eigenschappen

Naast de eigenschappen van het [Object Instance](#) heeft het object CompiledClipInstance de volgende eigenschappen:

Eigenschap	Beschrijving
<code>compiledClipInstance.accName</code>	Een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid.
<code>compiledClipInstance.actionScript</code>	Een tekenreeks die het ActionScript voor deze instantie vertegenwoordigt; gelijk aan <code>symbolInstance.actionScript</code> .
<code>compiledClipInstance.description</code>	Een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid.
<code>compiledClipInstance.forceSimple</code>	Een Booleaanse waarde die de toegankelijkheid van de onderliggende items van het object in- en uitschakelt.
<code>compiledClipInstance.shortcut</code>	Een tekenreeks die gelijk is aan het veld Sneltoets in het deelvenster Toegankelijkheid.
<code>compiledClipInstance.silent</code>	Een Booleaanse waarde die de toegankelijkheid van het object in- en uitschakelt; gelijk aan de omgekeerde logica van de instelling Object toegankelijk maken in het deelvenster Toegankelijkheid.
<code>compiledClipInstance.tabIndex</code>	Een geheel getal dat gelijk is aan het veld Tabindex in het deelvenster Toegankelijkheid.

## compiledClipInstance.accName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`compiledClipInstance.accName`

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid. Schermlezers identificeren objecten door de naam hardop voor te lezen.

### Voorbeeld

In het volgende voorbeeld wordt de toegankelijkheidsnaam van het eerste geselecteerde object opgehaald en ingesteld:

```
// Get the name of the object.  
var theName = fl.getDocumentDOM().selection[0].accName;  
// Set the name of the object.  
fl.getDocumentDOM().selection[0].accName = 'Home Button';
```

## compiledClipInstance.actionScript

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
compiledClipInstance.actionScript
```

### Beschrijving

Eigenschap; een tekenreeks die het ActionScript voor deze instantie vertegenwoordigt; gelijk aan [symbolInstance.actionScript](#).

### Voorbeeld

Met de volgende code wordt ActionScript toegewezen aan opgegeven elementen:

```
// Assign some ActionScript to a specified Button compiled clip instance.  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0]  
    .actionScript = "on(click) {trace('button is clicked');}";  
// Assign some ActionScript to the currently selected Button compiled clip instance.  
fl.getDocumentDOM().selection[0].actionScript =  
    "on(click) {trace('button is clicked');}";
```

## compiledClipInstance.description

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
compiledClipInstance.description
```

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid. De beschrijving wordt gelezen door de schermlezer.

### Voorbeeld

In het volgende voorbeeld wordt het ophalen en instellen van de eigenschap `description` gedemonstreerd:

```
// Get the description of the current selection.  
var theDescription = fl.getDocumentDOM().selection[0].description;  
// Set the description of the current selection.  
fl.getDocumentDOM().selection[0].description =  
    "This is compiled clip number 1";
```

## compiledClipInstance.forceSimple

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
compiledClipInstance.forceSimple
```

### Beschrijving

Eigenschap; een Booleaanse waarde die de toegankelijkheid van de onderliggende items van het object in- en uitschakelt. Dit is gelijk aan de omgekeerde logica van de instelling `Onderliggende objecten toegankelijk maken` in het deelvenster `Toegankelijkheid`. `forceSimple` is `true`, is gelijk aan de optie `Onderliggende objecten toegankelijk maken` die is uitgeschakeld. `forceSimple` is `false`, is gelijk aan de optie `Onderliggende objecten toegankelijk maken` die is ingeschakeld.

### Voorbeeld

In het volgende voorbeeld wordt het ophalen en instellen van de eigenschap `forceSimple` gedemonstreerd:

```
// Query if the children of the object are accessible.  
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;  
// Allow the children of the object to be accessible.  
fl.getDocumentDOM().selection[0].forceSimple = false;
```

## compiledClipInstance.shortcut

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
compiledClipInstance.shortcut
```

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld `Sneltoets` in het deelvenster `Toegankelijkheid`. De sneltoets wordt gelezen door de schermlezer. Deze eigenschap is niet beschikbaar voor dynamische tekstvelden.

### Voorbeeld

In het volgende voorbeeld wordt het ophalen en instellen van de eigenschap `shortcut` gedemonstreerd:

```
// Get the shortcut key of the object.  
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;  
// Set the shortcut key of the object.  
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+I";
```

## compiledClipInstance.silent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
compiledClipInstance.silent
```

### Beschrijving

Eigenschap; een Booleaanse waarde die de toegankelijkheid van het object in- en uitschakelt; gelijk aan de omgekeerde logica van de instelling Object toegankelijk maken in het deelvenster Toegankelijkheid. Dit betekent dat als `silenttrue` is, de optie Object toegankelijk maken is uitgeschakeld. Als `silentfalse` is, is de optie Object toegankelijk maken ingeschakeld.

### Voorbeeld

In het volgende voorbeeld wordt het ophalen en instellen van de eigenschap `silent` gedemonstreerd:

```
// Query if the object is accessible.  
var isSilent = fl.getDocumentDOM().selection[0].silent;  
// Set the object to be accessible.  
fl.getDocumentDOM().selection[0].silent = false;
```

## compiledClipInstance.tabIndex

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
compiledClipInstance.tabIndex
```

### Beschrijving

Eigenschap; een geheel getal dat gelijk is aan het veld Tabindex in het deelvenster Toegankelijkheid. Hiermee wordt een tabvolgorde gemaakt waarin objecten worden benaderd wanneer de gebruiker op de Tab-toets drukt.

### Voorbeeld

In het volgende voorbeeld wordt het ophalen en instellen van de eigenschap `tabIndex` gedemonstreerd:

```
// Get the tabIndex of the object.  
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;  
// Set the tabIndex of the object.  
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

# Hoofdstuk 7: Object `compilerErrors`

## Beschikbaarheid

Flash CS3 Professional.

## Beschrijving

Het object `compilerErrors`, dat het deelvenster Compilerfouten vertegenwoordigt, is een eigenschap van het object `flash (fl)` en is toegankelijk via `fl.compilerErrors` (zie [Object flash \(fl\)](#)).

## Overzicht van methoden

Voor het object `compilerErrors` kunnen de volgende methoden worden gebruikt:

Methoden	Beschrijving
<code>compilerErrors.clear()</code>	Hiermee wordt de inhoud van het deelvenster Compilerfouten gewist.
<code>compilerErrors.save()</code>	Hiermee wordt de inhoud van het deelvenster Compilerfouten in een lokaal tekstbestand opgeslagen.

## `compilerErrors.clear()`

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
compilerErrors.clear()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methoden; hiermee wordt de inhoud van het deelvenster Compilerfouten gewist.

### Voorbeeld

In het volgende voorbeeld wordt de inhoud van het deelvenster Compilerfouten gewist:

```
fl.compilerErrors.clear();
```

### Zie ook

[compilerErrors.save\(\)](#)

## `compilerErrors.save()`

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
compilerErrors.save(fileURI [, bAppendToFile [, bUseSystemEncoding]])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, dat de bestandsnaam van het opgeslagen bestand aangeeft. Als *fileURI* al bestaat en u niet de waarde `true` hebt opgegeven voor *bAppendToFile*, wordt *fileURI* zonder waarschuwing overschreven.

**bAppendToFile** Een optionele Boole-waarde die aangeeft of de inhoud van het deelvenster Compilerfouten wel (`true`) of niet (`false`) moet worden toegevoegd aan *fileURI*. De standaardwaarde is `false`.

**bUseSystemEncoding** Een optionele Boole-waarde die aangeeft of de tekst uit het deelvenster Compilerfouten moet worden opgeslagen met de systeemcodering. Als deze waarde `false` is (de standaardwaarde), wordt de tekst uit het deelvenster Compilerfouten opgeslagen met UTF-8-codering, met markeringstekens voor de volgorde van bytes aan het begin van de tekst. De standaardwaarde is `false`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de inhoud van het deelvenster Compilerfouten in een lokaal tekstbestand opgeslagen.

### Voorbeeld

In het volgende voorbeeld wordt de inhoud van het deelvenster Compilerfouten opgeslagen in het bestand `errors.log` in de map `C:\tests`:

```
fl.compilerErrors.save("file:///c:/tests/errors.log");
```

### Zie ook

[compilerErrors.clear\(\)](#)

# Hoofdstuk 8: Object ComponentInstance

**Overerving** [Object Element](#) > [Object Instance](#) > [Object SymbolInstance](#) > Object ComponentInstance

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object ComponentInstance is een subklasse van het object SymbolInstance en vertegenwoordigt een bitmap in een frame (zie [Object SymbolInstance](#)).

## Overzicht van eigenschappen

Naast alle eigenschappen van het [Object SymbolInstance](#) heeft het object ComponentInstance de volgende eigenschap:

Eigenschap	Beschrijving
<code>componentInstance.parameters</code>	Alleen-lezen; een array met ActionScript 2.0-eigenschappen die kunnen worden benaderd vanuit de Eigenschapcontrole voor componenten.

## componentInstance.parameters

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`componentInstance.parameters`

### Beschrijving

Alleen-lezen eigenschap; een array met ActionScript 2.0-eigenschappen die kunnen worden benaderd vanuit de Eigenschapcontrole voor componenten. Zie [Object Parameter](#).

### Voorbeeld

In het volgende voorbeeld wordt het ophalen en instellen van de eigenschap `parameters` gedemonstreerd:

```
var parms = fl.getDocumentDOM().selection[0].parameters;
parms[0].value = "some value";
```

### Zie ook

[Object Parameter](#)

# Hoofdstuk 9: Object componentsPanel

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object `componentsPanel` dat het deelvenster `Components` vertegenwoordigt, is een eigenschap van het object `flash (fl)` die kan worden benaderd via `fl.componentsPanel` (zie [Object flash \(fl\)](#)).

## Overzicht van methoden

U kunt de volgende methoden gebruiken met het object `componentsPanel`:

Methoden	Beschrijving
<code>componentsPanel.addItemToDocument()</code>	Hiermee wordt de opgegeven component op de opgegeven positie toegevoegd aan het document.
<code>componentsPanel.reload()</code>	Hiermee wordt de lijst met componenten in het deelvenster <code>Componenten</code> vernieuwd.

## `componentsPanel.addItemToDocument()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
componentsPanel.addItemToDocument(position, categoryName, componentName)
```

### Parameters

**position** Een punt (bijvoorbeeld `{x:0, y:100}`) dat de locatie vastlegt waarop het component moet worden toegevoegd. Geef *position* op ten opzichte van het middelpunt van de component, niet ten opzichte van het registratiepunt van de component (ook *oorsprongpunt* of *nulpunt*).

**categoryName** Een tekenreeks waarmee de naam van de componentcategorie wordt opgegeven (bijvoorbeeld "Data"). De geldige categorienamen worden weergegeven in het deelvenster `Componenten`.

**componentName** Een tekenreeks waarmee de naam wordt opgegeven van de component in de opgegeven categorie (bijvoorbeeld "WebServiceConnector"). De geldige componentnamen worden weergegeven in het deelvenster `Componenten`.

### Geretourneerde waarde

Niets.

### Beschrijving

Hiermee wordt de opgegeven component op de opgegeven positie toegevoegd aan het document.



### Voorbeeld

In de volgende voorbeelden worden enkele voorbeelden van het gebruik van deze methode getoond:

```
fl.componentsPanel.addItemToDocument({x:0, y:0}, "User Interface", "CheckBox");  
fl.componentsPanel.addItemToDocument({x:0, y:100}, "Data", "WebServiceConnector");  
fl.componentsPanel.addItemToDocument({x:0, y:200}, "User Interface", "Button");
```

## componentsPanel.reload()

### Beschikbaarheid

Flash 8.

### Gebruik

```
componentsPanel.reload()
```

### Parameters

Geen.

### Geretoureerde waarde

Een Booleaanse waarde: `true` als de lijst in het deelvenster Componenten is vernieuwd, anders `false`.

### Beschrijving

Methode; hiermee wordt de lijst met componenten in het deelvenster Componenten vernieuwd.

### Voorbeeld

In het volgende voorbeeld wordt het deelvenster Componenten vernieuwd.

```
fl.componentsPanel.reload();
```

# Hoofdstuk 10: Object Contour

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Een object Contour vertegenwoordigt een gesloten pad van halve randen op de omtrek van een vorm.

## Overzicht van methoden

U kunt de volgende methode gebruiken met het object Contour:

Methode	Beschrijving
<code>contour.getHalfEdge()</code>	Hiermee wordt een <a href="#">Object HalfEdge</a> geretourneerd op de contour van de selectie.

## Overzicht van eigenschappen

U kunt de volgende eigenschappen gebruiken met het object Contour:

Eigenschap	Beschrijving
<code>contour.fill</code>	Een <a href="#">Object Fill</a> .
<code>contour.interior</code>	Alleen-lezen: de waarde is <code>true</code> als de contour een gebied omsluit, anders <code>false</code> .
<code>contour.orientation</code>	Alleen-lezen; een geheel getal dat de oriëntatie van de contour aangeeft.

## contour.fill

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`contour.fill`

### Beschrijving

Eigenschap: een [Object Fill](#).

### Voorbeeld

Ervan uitgaande dat u een contour hebt met een geselecteerde vulling, wordt in het volgende voorbeeld de vulkleur van de contour weergegeven in het deelvenster Uitvoer:

```
var insideContour = fl.getDocumentDOM().selection[0].contours[1];
var insideFill = insideContour.fill;
fl.trace(insideFill.color);
```

## contour.getHalfEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
contour.getHalfEdge()
```

### Parameters

Geen.

### Geretourneerde waarde

Een [Object HalfEdge](#).

### Beschrijving

Methode; hiermee wordt een [Object HalfEdge](#) geretourneerd op de contour van de selectie.

### Voorbeeld

In dit voorbeeld worden alle contouren van de geselecteerde vorm doorlopen en de coördinaten van de hoekpunten weergegeven in het deelvenster Uitvoer:

```
// with a shape selected

var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;
var contourCount = 0;
for (i=0;i<contourArray.length;i++)
{
    var contour = contourArray[i];
    contourCount++;
    var he = contour.getHalfEdge();

    var iStart = he.id;
    var id = 0;
    while (id != iStart)
    {
        // Get the next vertex.
        var vrt = he.getVertex();

        var x = vrt.x;
        var y = vrt.y;
        fl.trace("vrt: " + x + ", " + y);

        he = he.getNext();
        id = he.id;
    }
}
elt.endEdit();
```

## contour.interior

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`contour.interior`

### Beschrijving

Alleen-lezen eigenschap; de waarde is `true` als de contour een gebied omsluit, anders `false`.

### Voorbeeld

In dit voorbeeld worden alle contouren van de geselecteerde vorm doorlopen en wordt de waarde van de eigenschap `interior` voor elke contour weergegeven in het deelvenster Uitvoer:

```
var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;

var contourCount = 0;
for (i=0;i<contourArray.length;i++) {
    var contour = contourArray[i];
    fl.trace("Next Contour, interior:" + contour.interior );
    contourCount++;
}
elt.endEdit();
```

## contour.orientation

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`contour.orientation`

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat de oriëntatie van de contour aangeeft. De waarde van het gehele getal is -1 als de oriëntatie linksom is, 1 als de oriëntatie rechtsom is en 0 als het een contour is zonder gebied.

### Voorbeeld

In dit voorbeeld worden alle contouren in de geselecteerde vorm doorlopen en wordt de waarde van de eigenschap `interior` voor elke contour getoond in het deelvenster Uitvoer:

```
var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;

var contourCount = 0;
for (i=0;i<contourArray.length;i++) {
    var contour = contourArray[i];
    fl.trace("Next Contour, orientation:" + contour.orientation);
    contourCount++;
}
elt.endEdit();
```

# Hoofdstuk 11: Object Document

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Document vertegenwoordigt het werkgebied. Dat wil zeggen, dat alleen FLA-bestanden als documenten worden beschouwd. Gebruik `fl.getDocumentDOM()` om het object Document voor het huidige document te retourneren.

## Overzicht van methoden

U kunt de volgende methoden gebruiken met het object Document:

Methoden	Beschrijving
<code>document.addDataToDocument()</code>	Hiermee worden opgegeven gegevens opgeslagen bij een document.
<code>document.addDataToSelection()</code>	Hiermee worden opgegeven gegevens opgeslagen met een of meer geselecteerde objecten.
<code>document.addFilter()</code>	Hiermee worden filters toegepast op de geselecteerde objecten.
<code>document.addItem()</code>	Hiermee wordt een item vanuit een geopend document of een bibliotheek toegevoegd aan het opgegeven object Document.
<code>document.addNewLine()</code>	Hiermee wordt een nieuw pad tussen twee punten toegevoegd.
<code>document.addNewOval()</code>	Hiermee wordt een nieuw object Oval toegevoegd in het opgegeven kader.
<code>document.addNewPrimitiveOval()</code>	Hiermee wordt een nieuwe primitieve ovaal toegevoegd die past binnen de opgegeven grenzen.
<code>document.addNewPrimitiveRectangle()</code>	Hiermee wordt een nieuwe primitieve rechthoek toegevoegd die past binnen de opgegeven grenzen.
<code>document.addNewPublishProfile()</code>	Hiermee wordt een nieuw publicatieprofiel toegevoegd en geactiveerd.
<code>document.addNewRectangle()</code>	Hiermee wordt een nieuwe rechthoek of afgeronde rechthoek toegevoegd die past binnen de opgegeven grenzen.
<code>document.addNewScene()</code>	Hiermee wordt een nieuwe scène (Object Timeline) toegevoegd achter de geselecteerde scène en wordt de nieuwe scène geselecteerd.
<code>document.addNewText()</code>	Hiermee wordt een nieuw leeg tekstveld ingevoegd.
<code>document.align()</code>	Hiermee wordt de selectie uitgelijnd.
<code>document.allowScreens()</code>	Gebruik deze methode alvorens de eigenschap " <code>document.screenOutline</code> " op pagina 140 te gebruiken.
<code>document.arrange()</code>	Hiermee wordt de selectie in het werkgebied gerangschikt.
<code>document.breakApart()</code>	Hiermee wordt de huidige selectie gesplitst.

Method	Beschrijving
<code>document.canEditSymbol()</code>	Hiermee wordt aangegeven of het menu en de functionaliteit Symbolen bewerken zijn ingeschakeld.
<code>document.canRevert()</code>	Hiermee wordt bepaald of u de methode <code>document.revert()</code> of <code>fl.revertDocument()</code> kunt gebruiken.
<code>document.canTestMovie()</code>	Hiermee wordt bepaald of u de methode <code>document.testMovie()</code> kunt gebruiken.
<code>document.canTestScene()</code>	Hiermee wordt bepaald of u de methode <code>document.testScene()</code> kunt gebruiken.
<code>document.changeFilterOrder()</code>	Hiermee wordt de index van het filter in de lijst met filters gewijzigd.
<code>document.clipCopy()</code>	Hiermee wordt de huidige selectie vanuit het document naar het klembord gekopieerd.
<code>document.clipCut()</code>	Hiermee wordt de huidige selectie uit het document geknipt en naar het klembord geschreven.
<code>document.clipPaste()</code>	Hiermee wordt de inhoud van het klembord in het document geplakt.
<code>document.close()</code>	Hiermee wordt het opgegeven document gesloten.
<code>document.convertLinesToFills()</code>	Hiermee worden lijnen omgezet in vullingen bij geselecteerde objecten.
<code>document.convertToSymbol()</code>	Hiermee worden een of meer geselecteerde items in het werkgebied geconverteerd naar een nieuw symbool.
<code>document.crop()</code>	Hiermee wordt het bovenste geselecteerde tekenobject gebruikt om alle geselecteerde tekenobjecten eronder bij te snijden.
<code>document.debugMovie()</code>	Hiermee wordt een foutopsporingssessie voor het document gestart.
<code>document.deleteEnvelope()</code>	Hiermee wordt de envelop (het kader dat een of meer objecten bevat) van het geselecteerde object verwijderd.
<code>document.deletePublishProfile()</code>	Hiermee wordt het huidige actieve profiel verwijderd als er meerdere zijn.
<code>document.deleteScene()</code>	Hiermee wordt de huidige scène (Object Timeline) verwijderd en wordt de volgende scène ingesteld als het huidige object Timeline, als de verwijderde scène niet de laatste scène was.
<code>document.deleteSelection()</code>	Hiermee wordt de huidige selectie verwijderd uit het werkgebied.
<code>document.disableAllFilters()</code>	Hiermee worden alle filters voor de geselecteerde objecten uitgeschakeld.
<code>document.disableFilter()</code>	Hiermee wordt het opgegeven filter in de lijst met filters uitgeschakeld.
<code>document.disableOtherFilters()</code>	Hiermee worden alle filters uitgeschakeld behalve het filter op de opgegeven positie in de lijst met filters.
<code>document.distribute()</code>	Hiermee wordt de selectie gedistribueerd.
<code>document.distributeToLayers()</code>	Hiermee wordt de huidige selectie verdeeld over lagen. Het effect is gelijk aan de opdracht Over lagen verdelen.

Method	Beschrijving
<code>document.documentHasData()</code>	Hiermee wordt gecontroleerd of het document permanente gegevens met de opgegeven naam bevat.
<code>document.duplicatePublishProfile()</code>	Hiermee wordt het huidige actieve profiel gedupliceerd en krijgt de gedupliceerde versie de focus.
<code>document.duplicateScene()</code>	Hiermee wordt een kopie gemaakt van de geselecteerde scène, waarbij de nieuwe scène een unieke naam krijgt en wordt ingesteld als de huidige scène.
<code>document.duplicateSelection()</code>	Hiermee wordt de selectie in het werkgebied gedupliceerd.
<code>document.editScene()</code>	Hiermee wordt de opgegeven scène geselecteerd voor bewerken.
<code>document.enableAllFilters()</code>	Hiermee worden alle filters in de lijst met filters ingeschakeld voor een of meer geselecteerde objecten.
<code>document.enableFilter()</code>	Hiermee wordt het opgegeven filter ingeschakeld voor een of meer geselecteerde objecten.
<code>document.enterEditMode()</code>	Hiermee wordt geschakeld naar de bewerkingsmodus die wordt opgegeven met de parameter.
<code>document.exitEditMode()</code>	Hiermee wordt de symboolbewerkingsmodus afgesloten en wordt het niveau boven de bewerkingsmodus geactiveerd.
<code>document.exportPNG()</code>	Hiermee wordt het document geëxporteerd als een of meer PNG-bestanden.
<code>document.exportPublishProfile()</code>	Hiermee wordt het huidige actieve profiel geëxporteerd naar een XML-bestand.
<code>document.exportPublishProfileString()</code>	Hiermee wordt een tekenreeks geretourneerd met het opgegeven profiel in XML-indeling.
<code>document.exportSWF()</code>	Hiermee wordt het document geëxporteerd in de Flash SWF-indeling.
<code>document.getAlignToDocument()</code>	Identiek aan het ophalen van de waarde van de knop Met werkgebied in het deelvenster Uittlijnen.
<code>document.getBlendMode()</code>	Hiermee wordt een tekenreeks geretourneerd waarmee de overvloedmodus voor een of meer geselecteerde objecten wordt opgegeven.
<code>document.getCustomFill()</code>	Hiermee wordt het object Fill opgehaald van de geselecteerde vorm of het deelvenster Gereedschappen en Eigenschapcontrole, indien opgegeven.
<code>document.getCustomStroke()</code>	Hiermee wordt het object Stroke opgehaald van de geselecteerde vorm of het deelvenster Gereedschappen en Eigenschapcontrole, indien opgegeven.
<code>document.getDataFromDocument()</code>	Hiermee wordt de waarde van de opgegeven gegevens opgehaald.
<code>document.getElementProperty()</code>	Hiermee wordt de opgegeven eigenschap <code>Element</code> voor de huidige selectie opgehaald.
<code>document.getElementTextAttr()</code>	Hiermee wordt een specifieke eigenschap <code>TextAttrs</code> van de geselecteerde tekstobjecten opgehaald.
<code>document.getFilters()</code>	Hiermee wordt een array geretourneerd die de lijst bevat met filters die zijn toegepast op een of meer geselecteerde objecten.



Method	Beschrijving
<code>document.getMetadata()</code>	Hiermee wordt een tekenreeks geretourneerd met de XML-metagegevens die aan het document zijn gekoppeld.
<code>document.getMobileSettings()</code>	Hiermee wordt de tekenreeks geretourneerd die aan <code>document.setMobileSettings()</code> is doorgegeven.
<code>document.getPlayerVersion()</code>	Hiermee wordt een tekenreeks geretourneerd die de doelversie van Flash Player voor het opgegeven document vertegenwoordigt.
<code>document.getSelectionRect()</code>	Hiermee wordt het kader van de huidige selectie opgehaald.
<code>document.getTextString()</code>	Hiermee wordt de geselecteerde tekst opgehaald.
<code>document.getTimeline()</code>	Hiermee wordt het huidige <a href="#">Object Timeline</a> in het document opgehaald.
<code>document.getTransformationPoint()</code>	Hiermee wordt de locatie opgehaald van het transformatiepunt van de huidige selectie.
<code>document.group()</code>	Hiermee wordt de huidige selectie omgezet in een groep.
<code>document.importFile()</code>	Hiermee wordt een bestand geïmporteerd in het document.
<code>document.importPublishProfile()</code>	Hiermee wordt een profiel uit een bestand geïmporteerd.
<code>document.importPublishProfileString()</code>	Hiermee wordt een XML-tekenreeks die een publicatieprofiel bevat geïmporteerd en ingesteld als het huidige profiel.
<code>document.importSWF()</code>	Hiermee wordt een SWF-bestand geïmporteerd in het document.
<code>document.intersect()</code>	Hiermee wordt een doorsnede van een object Drawing gemaakt van alle geselecteerde objecten Drawing.
<code>document.loadCuepointXML()</code>	Hiermee wordt een XML-actiepuntbodybestand geladen.
<code>document.match()</code>	Hiermee worden alle geselecteerde objecten even groot gemaakt.
<code>document.mouseClick()</code>	Hiermee wordt een muisklik uitgevoerd met het gereedschap Selectie.
<code>document.mouseDoubleClick()</code>	Hiermee wordt een dubbele muisklik uitgevoerd met het gereedschap Selectie.
<code>document.moveSelectedBezierPointsBy()</code>	Als de selectie ten minste één pad bevat waarop ten minste één Bézier-punt is geselecteerd, worden met deze methode alle geselecteerde Bézier-punten op alle geselecteerde paden met de opgegeven hoeveelheid verplaatst.
<code>document.moveSelectionBy()</code>	Hiermee worden geselecteerde objecten over een bepaalde afstand verplaatst.
<code>document.optimizeCurves()</code>	Hiermee wordt het vloeiend maken voor de huidige selectie geoptimaliseerd, waarbij het optimum desgewenst in meerdere keren kan worden bereikt; gelijk aan het selecteren van Wijzigen > Vorm > Optimaliseren.
<code>document.publish()</code>	Hiermee wordt het document gepubliceerd volgens de actieve publicatie-instellingen (Bestand > Publicatie-instellingen); gelijk aan Bestand > Publiceren.
<code>document.punch()</code>	Hiermee wordt het bovenste geselecteerde tekenobject geperforeerd door alle onderliggende geselecteerde tekenobjecten.

Method	Beschrijving
<code>document.removeAllFilters()</code>	Hiermee worden alle filters van een of meer geselecteerde objecten verwijderd.
<code>document.removeDataFromDocument()</code>	Hiermee worden permanente gegevens met de opgegeven naam die aan het document zijn gekoppeld, verwijderd.
<code>document.removeDataFromSelection()</code>	Hiermee worden permanente gegevens met de opgegeven naam die aan de selectie zijn gekoppeld, verwijderd.
<code>document.removeFilter()</code>	Hiermee wordt het opgegeven filter verwijderd uit de lijst met filters van een of meer geselecteerde objecten.
<code>document.renamePublishProfile()</code>	Hiermee wordt de naam van het huidige profiel gewijzigd.
<code>document.renameScene()</code>	Hiermee wordt de naam gewijzigd van de geselecteerde scène in het deelvenster Scènes.
<code>document.reorderScene()</code>	Hiermee wordt de opgegeven scène vóór een andere opgegeven scène geplaatst.
<code>document.resetOvalObject()</code>	Hiermee worden alle waarden in Eigenschapcontrole ingesteld op standaardinstellingen voor het object Oval.
<code>document.resetRectangleObject()</code>	Hiermee worden alle waarden in Eigenschapcontrole ingesteld op standaardinstellingen voor het object Rectangle.
<code>document.resetTransformation()</code>	Hiermee wordt de transformatiematrix opnieuw ingesteld; gelijk aan Wijzigen > Transformeren > Transformatie verwijderen.
<code>document.revert()</code>	Hiermee keert u terug naar de laatst opgeslagen versie van het opgegeven document; gelijk aan Bestand > Terugkeren.
<code>document.rotate3DSelection()</code>	Hiermee wordt een 3D-rotatie toegepast op de selectie.
<code>document.rotateSelection()</code>	Hiermee wordt de selectie geroteerd met een opgegeven aantal graden.
<code>document.save()</code>	Hiermee wordt het document opgeslagen op de standaardlocatie; gelijk aan Bestand > Opslaan.
<code>document.saveAndCompact()</code>	Hiermee wordt het bestand opgeslagen en gecomprimeerd; gelijk aan Bestand > Opslaan en comprimeren.
<code>document.scaleSelection()</code>	Hiermee wordt de selectie met een opgegeven factor geschaald; gelijk aan het gebruik van het gereedschap Vrije transformatie om het object te schalen.
<code>document.selectAll()</code>	Hiermee worden alle items in het werkgebied geselecteerd; gelijk aan drukken op Control+A (Windows) of Command+A (Macintosh) of de opdracht Bewerken > Alles selecteren.
<code>document.selectNone()</code>	Hiermee wordt de selectie van geselecteerde items opgeheven.
<code>document.setAlignToDocument()</code>	Hiermee worden de voorkeuren voor <code>document.align()</code> , <code>document.distribute()</code> , <code>document.match()</code> en <code>document.space()</code> voor het document ingesteld; gelijk aan het inschakelen van de knop Op werkgebied in het deelvenster Uitlijnen.
<code>document.setBlendMode()</code>	Hiermee wordt de overvloeimodus voor de geselecteerde objecten ingesteld.

Methode	Beschrijving
<code>document.setCustomFill()</code>	Hiermee worden de vulinstellingen ingesteld voor het deelvenster Gereedschappen, Eigenschapcontrole en alle geselecteerde vormen.
<code>document.setCustomStroke()</code>	Hiermee worden de streekinstellingen ingesteld voor het deelvenster Gereedschappen, Eigenschapcontrole en alle geselecteerde vormen.
<code>document.setElementProperty()</code>	Hiermee wordt de opgegeven eigenschap <code>Element</code> ingesteld voor een of meer geselecteerde objecten in het document.
<code>document.setElementTextAttr()</code>	Hiermee wordt de opgegeven eigenschap <code>TextAttrs</code> van de geselecteerde tekstitems ingesteld op de opgegeven waarde.
<code>document.setFillColor()</code>	Hiermee wordt de vulkleur van de selectie gewijzigd in de opgegeven kleur.
<code>document.setFilterProperty()</code>	Hiermee wordt een opgegeven filtereigenschap ingesteld voor de geselecteerde objecten.
<code>document.setFilters()</code>	Hiermee worden filters toegepast op de geselecteerde objecten.
<code>document.setInstanceAlpha()</code>	Hiermee wordt de dekking van de instantie ingesteld.
<code>document.setInstanceBrightness()</code>	Hiermee wordt de helderheid van de instantie ingesteld.
<code>document.setInstanceTint()</code>	Hiermee wordt de tint van de instantie ingesteld.
<code>document.setMetadata()</code>	Hiermee worden de XML-metagegevens voor het opgegeven document ingesteld, waarbij alle bestaande metagegevens worden overschreven.
<code>document.setMobileSettings()</code>	Hiermee wordt de waarde van een tekenreeks met XML-instellingen in een mobiel FLA-bestand ingesteld.
<code>document.setOvalObjectProperty()</code>	Hiermee wordt een waarde voor een opgegeven eigenschap van primitieve objecten <code>Oval</code> opgegeven.
<code>document.setPlayerVersion()</code>	Hiermee wordt de doelversie van Flash Player voor het opgegeven document ingesteld.
<code>document.setRectangleObjectProperty()</code>	Hiermee wordt een waarde voor een opgegeven eigenschap van primitieve objecten <code>Rectangle</code> opgegeven.
<code>document.setSelectionBounds()</code>	Hiermee wordt in één enkele bewerking de selectie verplaatst en het formaat gewijzigd.
<code>document.setSelectionRect()</code>	Hiermee wordt een rechthoekig selectiekader getekend relatief ten opzichte van het werkgebied, met de opgegeven coördinaten.
<code>document.setStageVanishingPoint()</code>	Hiermee geeft u het verdwijnpunt voor het weergeven van 3D-objecten op.
<code>document.setStageViewAngle()</code>	Hiermee geeft u de perspectiefhoek voor het weergeven van 3D-objecten op.
<code>document.setStroke()</code>	Hiermee worden de kleur, breedte en stijl van de geselecteerde streken ingesteld.
<code>document.setStrokeColor()</code>	Hiermee wordt de streekkleur van de selectie gewijzigd in de opgegeven kleur.
<code>document.setStrokeSize()</code>	Hiermee wordt de streekgrootte van de selectie gewijzigd in de opgegeven grootte.

Methode	Beschrijving
<code>document.setStrokeStyle()</code>	Hiermee wordt de strekstijl van de selectie gewijzigd in de opgegeven stijl.
<code>document.setTextRectangle()</code>	Hiermee wordt het kader voor het geselecteerde tekstitem gewijzigd in de opgegeven grootte.
<code>document.setTextSelection()</code>	Hiermee wordt in het huidige tekstveld de tekst geselecteerd die wordt bepaald door de waarden van <i>startIndex</i> en <i>endIndex</i> .
<code>document.setTextString()</code>	Hiermee wordt een tekenreeks ingevoegd.
<code>document.setTransformationPoint()</code>	Hiermee wordt het transformatiepunt van de huidige selectie verplaatst.
<code>document.skewSelection()</code>	Hiermee wordt de selectie een bepaald aantal graden scheefgetrokken.
<code>document.smoothSelection()</code>	Hiermee wordt de curve van elke geselecteerde contourvulling of gekromde lijn vloeiend gemaakt.
<code>document.space()</code>	Hiermee worden de objecten in de selectie gelijkmatig verdeeld.
<code>document.straightenSelection()</code>	Hiermee worden de geselecteerde streken rechtgetrokken; gelijk aan de knop Rechttrekken in het deelvenster Gereedschappen.
<code>document.swapElement()</code>	Hiermee wordt de huidige selectie omgewisseld met de opgegeven selectie.
<code>document.swapStrokeAndFill()</code>	Hiermee worden de streek- en vulkleuren gewisseld.
<code>document.testMovie()</code>	Hiermee wordt de bewerking Film testen uitgevoerd op het document.
<code>document.testScene()</code>	Hiermee wordt de bewerking Scène testen uitgevoerd op de huidige scène van het document.
<code>document.traceBitmap()</code>	Hiermee wordt de bewerking Bitmap overtrekken uitgevoerd op de huidige selectie; gelijk aan Wijzigen > Bitmap > Bitmap overtrekken.
<code>document.transformSelection()</code>	Hiermee wordt een algemene transformatie op de huidige selectie uitgevoerd door het toepassen van de matrix die in de argumenten is opgegeven.
<code>document.translate3Dcenter()</code>	Hiermee wordt de XYZ-positie ingesteld ten opzichte waarvan de selectie wordt omgezet of geroteerd.
<code>document.translate3Dselection()</code>	Hiermee wordt een 3D-translatie toegepast op de selectie.
<code>document.unGroup()</code>	Hiermee wordt de groepering van de huidige selectie opgeheven.
<code>document.union()</code>	Hiermee worden alle geselecteerde vormen gecombineerd in een tekenobject.
<code>document.unlockAllElements()</code>	Hiermee worden alle vergrendelde elementen in het geselecteerde frame ontgrendeld.
<code>document.xmlPanel()</code>	Hiermee wordt een XMLUI-dialoogvenster weergegeven.

### Overzicht van eigenschappen

U kunt de volgende eigenschappen gebruiken met het object Document.

Eigenschap	Beschrijving
<code>document.accName</code>	Een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid.
<code>document.as3AutoDeclare</code>	Een Booleaanse waarde die aangeeft of de instanties die in het werkgebied worden geplaatst, automatisch worden toegevoegd aan tijdelijklassen die door de gebruiker zijn gedefinieerd.
<code>document.as3Dialect</code>	Een tekenreeks die het ActionScript 3.0-'dialect' beschrijft dat in het opgegeven document wordt gebruikt.
<code>document.as3ExportFrame</code>	Een geheel getal dat aangeeft naar welk frame ActionScript 3.0-classes worden geëxporteerd.
<code>document.as3StrictMode</code>	Een Booleaanse waarde die aangeeft of de ActionScript 3.0-compiler moet compileren terwijl de strikte modus is ingeschakeld of uitgeschakeld.
<code>document.as3WarningsMode</code>	Een Booleaanse waarde die aangeeft of de ActionScript 3.0-compiler moet compileren terwijl de waarschuwingsmodus is ingeschakeld of uitgeschakeld.
<code>document.asVersion</code>	Een geheel getal dat aangeeft welke versie van ActionScript in het opgegeven bestand wordt gebruikt.
<code>document.autoLabel</code>	Een Booleaanse waarde die gelijk is aan het selectievakje Automatisch label in het deelvenster Toegankelijkheid.
<code>document.backgroundColor</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat staat voor de achtergrondkleur.
<code>document.currentPublishProfile</code>	Een tekenreeks met de naam van het actieve publicatieprofiel voor het opgegeven document.
<code>document.currentTimeline</code>	Een geheel getal dat de index van de actieve tijdlijn aangeeft.
<code>document.description</code>	Een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid.
<code>document.docClass</code>	Hiermee wordt de ActionScript 3.0-klasse op hoofdniveau opgegeven die aan het document is gekoppeld.
<code>document.externalLibraryPath</code>	Een tekenreeks die een lijst bevat met items in het externe ActionScript 3.0-bibliotheekpad van het document, dat de locatie aangeeft van SWC-bestanden die worden gebruikt als gedeelde runtime bibliotheken.
<code>document.forceSimple</code>	Een Booleaanse waarde die aangeeft of de onderliggende items van het opgegeven object toegankelijk zijn.
<code>document.frameRate</code>	Een drijvende-kommawaarde die aangeeft hoeveel frames per seconde worden weergegeven wanneer het SWF-bestand wordt afgespeeld; de standaardwaarde is 12.
<code>document.height</code>	Een geheel getal dat de hoogte van het document (werkgebied) in pixels aangeeft.
<code>document.id</code>	Een uniek geheel getal (automatisch toegewezen) dat een document tijdens een Flash-sessie identificeert.
<code>document.library</code>	Alleen-lezen; het <a href="#">Object library</a> voor een document.
<code>document.libraryPath</code>	Een tekenreeks die een lijst bevat met items in het ActionScript 3.0-bibliotheekpad van het document, dat de locatie aangeeft van SWC-bestanden of mappen die SWC-bestanden bevatten.
<code>document.livePreview</code>	Een Booleaanse waarde waarmee wordt opgegeven of Live voorvertoning is ingeschakeld.

Eigenschap	Beschrijving
<code>document.name</code>	Alleen-lezen; een tekenreeks met de naam van een document (FLA-bestand).
<code>document.path</code>	Alleen-lezen; een tekenreeks met het pad van het document in een platformspecifieke notatie.
<code>document.pathURI</code>	Alleen-lezen; een tekenreeks, uitgedrukt als bestand:/// URI, met het pad van het document.
<code>document.publishProfiles</code>	Alleen-lezen; een array met de namen van de publicatieprofielen voor het document.
<code>document.screenOutline</code>	Alleen-lezen; het huidige object ScreenOutline voor het document. Het object ScreenOutline wordt in Flash Professional CS5 en latere versies niet meer gebruikt.
<code>document.selection</code>	Een array met de geselecteerde objecten in het document.
<code>document.silent</code>	Een Booleaanse waarde die aangeeft of het object toegankelijk is.
<code>document.sourcePath</code>	Een tekenreeks die een lijst bevat met items in het ActionScript 3.0-bronpad van het document, dat de locatie aangeeft van ActionScript-klassebestanden.
<code>document.timelines</code>	Alleen-lezen; een array met tijdlijnobjecten (zie <a href="#">Object Timeline</a> ).
<code>document.viewMatrix</code>	Alleen-lezen; een <a href="#">Object Matrix</a> .
<code>document.width</code>	Een geheel getal dat de breedte van het document (werkgebied) in pixels aangeeft.
<code>document.zoomFactor</code>	Hiermee wordt het zoompercentage van het werkgebied tijdens het ontwerpen opgegeven.

## document.accName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.accName`

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid. Schermlezers identificeren objecten door de naam hardop voor te lezen.

### Voorbeeld

In het volgende voorbeeld wordt de toegankelijkheidsnaam van het document ingesteld op "Main Movie":

```
fl.getDocumentDOM().accName = "Main Movie";
```

In het volgende voorbeeld wordt de toegankelijkheidsnaam van het document opgehaald:

```
fl.trace(fl.getDocumentDOM().accName);
```

## document.addDataToDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addDataToDocument(name, type, data)
```

### Parameters

**name** Een tekenreeks met de naam van de gegevens die moeten worden toegevoegd.

**type** Een tekenreeks waarmee het type wordt gedefinieerd van de gegevens die moeten worden toegevoegd. Acceptabele waarden zijn "integer", "integerArray", "double", "doubleArray", "string" en "byteArray".

**data** De waarde die moet worden toegevoegd. Welke typen geldig zijn, is afhankelijk van de parameter *type*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden opgegeven gegevens samen met een document opgeslagen. Gegevens worden naar het FLA-bestand geschreven en zijn beschikbaar voor JavaScript wanneer het bestand opnieuw wordt geopend.

### Voorbeeld

In het volgende voorbeeld wordt het gehele getal 12 toegevoegd aan het huidige document:

```
fl.getDocumentDOM().addDataToDocument("myData", "integer", 12);
```

In het volgende voorbeeld wordt de waarde van de gegevens met de naam "myData" geretourneerd en wordt het resultaat weergegeven in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().getDataFromDocument("myData"));
```

### Zie ook

[document.getDataFromDocument\(\)](#), [document.removeDataFromDocument\(\)](#)

## document.addDataToSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addDataToSelection(name, type, data)
```

### Parameters

**name** Een tekenreeks waarmee de naam van de permanente gegevens wordt opgegeven.

**type** Hiermee wordt het type van de gegevens gedefinieerd. Acceptabele waarden zijn "integer", "integerArray", "double", "doubleArray", "string" en "byteArray".

**data** De waarde die moet worden toegevoegd. Welke typen geldig zijn, is afhankelijk van de parameter *type*.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee worden opgegeven gegevens opgeslagen met een of meer geselecteerde objecten. Gegevens worden naar het FLA-bestand geschreven en zijn beschikbaar voor JavaScript wanneer het bestand opnieuw wordt geopend. Alleen symbolen en bitmaps ondersteunen permanente gegevens.

#### Voorbeeld

In het volgende voorbeeld wordt het gehele getal 12 toegevoegd aan het geselecteerde object:

```
fl.getDocumentDOM().addDataToSelection("myData", "integer", 12);
```

#### Zie ook

[document.removeDataFromSelection\(\)](#)

## document.addFilter()

#### Beschikbaarheid

Flash 8.

#### Gebruik

```
document.addFilter(filterName)
```

#### Parameters

**filterName** Een tekenreeks waarmee het filter wordt opgegeven dat wordt toegevoegd aan de lijst met filters en dat wordt ingeschakeld voor een of meer geselecteerde objecten. Acceptabele waarden zijn "adjustColorFilter", "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" en "gradientGlowFilter".

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een filter toegepast op de geselecteerde objecten en wordt het filter aan het einde van de lijst met filters geplaatst.

#### Voorbeeld

In het volgende voorbeeld wordt een gloedfilter toegepast op een of meer geselecteerde objecten:

```
fl.getDocumentDOM().addFilter("glowFilter");
```



### Zie ook

`document.changeFilterOrder()`, `document.disableFilter()`, `document.enableFilter()`,  
`document.getFilters()`, `document.removeFilter()`, `document.setBlendMode()`,  
`document.setFilterProperty()`

## document.addItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addItem(position, item)
```

### Parameters

**position** Een punt waarmee de *x*- en *y*-coördinaat worden opgegeven van de locatie waar het item moet worden toegevoegd. Hierbij wordt het midden van een symbool of de linkerbovenhoek van een bitmap of video gebruikt.

**item** Een object Item waarmee het item wordt opgegeven dat moet worden toegevoegd en de bibliotheek waaruit het moet worden toegevoegd (zie [Object Item](#)).

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee wordt een item vanuit een geopend document of een bibliotheek toegevoegd aan het opgegeven object Document.

### Voorbeeld

In het volgende voorbeeld wordt het eerste item uit de bibliotheek toegevoegd aan het eerste document op de opgegeven locatie voor het geselecteerde symbool of de geselecteerde bitmap of video:

```
var item = fl.documents[0].library.items[0];  
fl.documents[0].addItem({x:0,y:0}, item);
```

In het volgende voorbeeld wordt het symbool `myMovieClip` uit de bibliotheek van het huidige document toegevoegd aan het huidige document:

```
var itemIndex = fl.getDocumentDOM().library.findItemIndex("myMovieClip");  
var theItem = fl.getDocumentDOM().library.items[itemIndex];  
fl.getDocumentDOM().addItem({x:0,y:0}, theItem);
```

In het volgende voorbeeld wordt het symbool `myMovieClip` vanuit het tweede document in de documentenarray toegevoegd aan het derde document in de documentenarray:

```
var itemIndex = fl.documents[1].library.findItemIndex("myMovieClip");  
var theItem = fl.documents[1].library.items[itemIndex];  
fl.documents[2].addItem({x:0,y:0}, theItem);
```

## document.addNewLine()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addNewLine(startPoint, endpoint)
```

### Parameters

**startpoint** Een paar drijvende-kommawaarden waarmee de *x*- en *y*-coördinaat worden opgegeven van het beginpunt van de lijn.

**endpoint** Een paar drijvende-kommawaarden waarmee de *x*- en *y*-coördinaat worden opgegeven van het eindpunt van de lijn.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuw pad toegevoegd tussen twee punten. De methode gebruikt de huidige strekkenmerken van het document en voegt het pad toe aan het huidige frame en de huidige laag. Deze methode werkt hetzelfde als op het gereedschap Lijn klikken en een lijn tekenen.

### Voorbeeld

In het volgende voorbeeld wordt een lijn toegevoegd tussen het opgegeven beginpunt en eindpunt:

```
fl.getDocumentDOM().addNewLine({x:216.7, y:122.3}, {x:366.8, y:165.8});
```

## document.addNewOval()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addNewOval(boundingRectangle [, bSuppressFill [, bSuppressStroke ]])
```

### Parameters

**boundingRectangle** Een rechthoek waarmee de grenzen worden opgegeven van de ovaal die moet worden toegevoegd. Meer informatie over het formaat van *boundingRectangle* vindt u in [document.addNewRectangle\(\)](#).

**bSuppressFill** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de vorm zonder vulling gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

**bSuppressStroke** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de vorm zonder een streek gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuw object Oval toegevoegd in het opgegeven kader. Deze methode voert dezelfde bewerking uit als het gereedschap Ovaal. De methode gebruikt de huidige streek- en vulkenmerken van het document en voegt de ovaal toe aan het huidige frame en de huidige laag. Als zowel *bSuppressFill* als *bSuppressStroke* op `true` is ingesteld, heeft de methode geen effect.

### Voorbeeld

In het volgende voorbeeld wordt een nieuwe ovaal toegevoegd binnen de opgegeven coördinaten; de ovaal is 164 pixels breed en 178 pixels hoog:

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228});
```

In het volgende voorbeeld wordt de ovaal zonder vulling getekend:

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, true);
```

In het volgende voorbeeld wordt de ovaal zonder een streek getekend:

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, false, true);
```

### Zie ook

[document.addNewPrimitiveOval\(\)](#)

## document.addNewPrimitiveOval()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.addNewPrimitiveOval( boundingRectangle [, bSpupressFill [, bSuppressStroke ] ] )
```

### Parameters

**boundingRectangle** Een rechthoek die de grenzen bepaalt waarbinnen de nieuwe primitieve ovaal wordt toegevoegd. Zie [document.addNewRectangle\(\)](#) voor meer informatie over de notatie van *boundingRectangle*.

**bSuppressFill** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de ovaal zonder vulling gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

**bSuppressStroke** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de ovaal zonder een streek gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuwe primitieve ovaal toegevoegd die past binnen de opgegeven grenzen. Deze methode voert dezelfde bewerking uit als het gereedschap Primitieve ovaal. De primitieve ovaal gebruikt de huidige streek- en vulkenmerken van het document en wordt toegevoegd aan het huidige frame en de huidige laag. Als zowel *bSuppressFill* als *bSuppressStroke* op `true` is ingesteld, heeft de methode geen effect.

### Voorbeeld

In het volgende voorbeeld worden primitieve ovalen toegevoegd binnen de opgegeven coördinaten, met en zonder vulling en streek:

```
// Add an oval primitive with fill and stroke
fl.getDocumentDOM().addNewPrimitiveOval({left:0,top:0,right:100,bottom:100});
// Add an oval primitive without a fill
fl.getDocumentDOM().addNewPrimitiveOval({left:100,top:100,right:200,bottom:200}, true);
// Add an oval primitive without a stroke
fl.getDocumentDOM().addNewPrimitiveOval({left:200,top:200,right:300,bottom:300}, false, true);
```

### Zie ook

[document.addNewOval\(\)](#)

## document.addNewPrimitiveRectangle()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.addNewPrimitiveRectangle( boundingRectangle, roundness, [, bSuppressFill [,
bSuppressStroke ]] )
```

### Parameters

**rect** Een rechthoek die de grenzen bepaalt waarbinnen de nieuwe primitieve rechthoek wordt toegevoegd. Zie [document.addNewRectangle\(\)](#) voor meer informatie over de notatie van *boundingRectangle*.

**roundness** Een geheel getal in het bereik van 0 tot en met 999 dat het aantal punten aangeeft waarmee wordt bepaald hoeveel de hoeken moeten worden afgerond.

**bSuppressFill** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de rechthoek zonder vulling gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

**bSuppressStroke** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de rechthoek zonder een streek gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuwe primitieve rechthoek toegevoegd die past binnen de opgegeven grenzen. Deze methode voert dezelfde bewerking uit als het gereedschap Primitieve rechthoek. De primitieve rechthoek gebruikt de huidige streek- en vulkenmerken van het document en wordt toegevoegd aan het huidige frame en de huidige laag. Als zowel *bSuppressFill* als *bSuppressStroke* op `true` is ingesteld, heeft de methode geen effect.

### Voorbeeld

In het volgende voorbeeld worden primitieve rechthoeken toegevoegd binnen de opgegeven coördinaten, met en zonder vulling en streek en met verschillende rondingen:

```
// Add a rectangle primitive with fill and stroke
fl.getDocumentDOM().addNewPrimitiveRectangle({left:0,top:0,right:100,bottom:100}, 0);
// Add a rectangle primitive without a fill
fl.getDocumentDOM().addNewPrimitiveRectangle({left:100,top:100,right:200,bottom:200}, 20,
true);
// Add a rectangle primitive without a stroke
fl.getDocumentDOM().addNewPrimitiveRectangle({left:200,top:200,right:300,bottom:300},
50,false,true);
```

#### Zie ook

[document.addNewRectangle\(\)](#)

## document.addNewPublishProfile()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.addNewPublishProfile([profileName])
```

#### Parameters

**profileName** De unieke naam van het nieuwe profiel. Als u geen naam opgeeft, wordt een standaardnaam gebruikt. Deze parameter is optioneel.

#### Geretourneerde waarde

Een geheel getal dat de index is van het nieuwe profiel in de lijst met profielen. Retourneert -1 als er geen nieuw profiel kan worden gemaakt.

#### Beschrijving

Methode; hiermee wordt een nieuw publicatieprofiel toegevoegd en geactiveerd.

#### Voorbeeld

In het volgende voorbeeld wordt een nieuw publicatieprofiel met een standaardnaam toegevoegd en wordt de naam van het profiel weergegeven in het deelvenster Uitvoer:

```
fl.getDocumentDOM().addNewPublishProfile();
fl.outputPanel.trace(fl.getDocumentDOM().currentPublishProfile);
```

In het volgende voorbeeld wordt een nieuw publicatieprofiel met de naam "my profile" toegevoegd:

```
fl.getDocumentDOM().addNewPublishProfile("my profile");
```

#### Zie ook

[document.deletePublishProfile\(\)](#)

## document.addNewRectangle()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addNewRectangle(boundingRectangle, roundness  
    [, bSuppressFill [, bSuppressStroke]])
```

### Parameters

**boundingRectangle** Een rechthoek die de grenzen bepaalt waarbinnen de nieuwe rechthoek wordt toegevoegd, in de notatie {left:value1,top:value2,right:value3,bottom:value4}. De waarden `left` en `top` bepalen de locatie van de linkerbovenhoek (`left:0,top:0` is bijvoorbeeld de linkerbovenhoek van het werkgebied) en de waarden `right` en `bottom` bepalen de locatie van de rechteronderhoek. De breedte van de rechthoek is daarom het verschil tussen de waarden `left` en `right` en de hoogte van de rechthoek is het verschil tussen de waarden `top` en `bottom`.

Dit betekent dat niet alle grenzen van de rechthoek overeenkomen met de waarden die worden weergegeven in Eigenschapcontrole. De waarden `left` en `top` komen overeen met de waarden X respectievelijk Y in de Eigenschapcontrole. De waarden `right` en `bottom` komen echter niet overeen met de waarden W en H in de Eigenschapcontrole. Hieronder volgt als voorbeeld een rechthoek met de volgende grenzen:

```
{left:10,top:10,right:50,bottom:100}
```

Voor deze rechthoek worden de volgende waarden weergegeven in Eigenschapcontrole:

```
X = 10, Y = 10, W = 40, H = 90
```

**roundness** Een geheel getal tussen 0 en 999 dat de ronding van de hoeken aangeeft. De waarde wordt opgegeven als een aantal punten. Hoe groter de waarde, hoe groter de ronding.

**bSuppressFill** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de vorm zonder vulling gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

**bSuppressStroke** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de rechthoek zonder een strek gemaakt. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuwe primitieve rechthoek of afgeronde rechthoek toegevoegd die past binnen de opgegeven grenzen. Deze methode voert dezelfde bewerking uit als het gereedschap Rechthoek. De methode gebruikt de huidige streek- en vulkenmerken van het document en voegt de rechthoek toe aan het huidige frame en de huidige laag. Als zowel *bSuppressFill* als *bSuppressStroke* op `true` is ingesteld, heeft de methode geen effect.

### Voorbeeld

In het volgende voorbeeld wordt een nieuwe rechthoek zonder afgeronde hoeken toegevoegd binnen de opgegeven coördinaten; de rechthoek is 100 pixels breed en 100 pixels hoog:

```
fl.getDocumentDOM().addNewRectangle({left:0,top:0,right:100,bottom:100},0);
```

In het volgende voorbeeld wordt een nieuwe rechthoek zonder afgeronde hoeken en zonder vulling toegevoegd; de rechthoek is 100 pixels breed en 200 pixels hoog:

```
fl.getDocumentDOM().addNewRectangle({left:10,top:10,right:110,bottom:210},0,true);
```

In het volgende voorbeeld wordt een nieuwe rechthoek zonder afgeronde hoeken en zonder een streek toegevoegd; de rechthoek is 200 pixels breed en 100 pixels hoog:

```
fl.getDocumentDOM().addNewRectangle({left:20,top:20,right:220,bottom:120},0,false,true);
```

### Zie ook

[document.addNewPrimitiveRectangle\(\)](#)

## document.addNewScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.addNewScene([name])
```

### Parameters

**name** Hier geeft u de naam van de scène op. Als u geen naam opgeeft, wordt een nieuwe scène naam gegenereerd.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de scène is toegevoegd, anders `false`.

### Beschrijving

Methode; hiermee wordt een nieuwe scène ([Object Timeline](#)) toegevoegd achter de geselecteerde scène en wordt de nieuwe scène geselecteerd. Als de opgegeven scène naam al bestaat, wordt de scène niet toegevoegd en retourneert de methode een fout.

### Voorbeeld

In het volgende voorbeeld wordt een scène met de naam `myScene` toegevoegd achter de huidige scène in het huidige document. De waarde van de variabele `success` is `true` als de nieuwe scène is gemaakt, anders `false`.

```
var success = fl.getDocumentDOM().addNewScene("myScene");
```

In het volgende voorbeeld wordt een scène toegevoegd met de standaardnaamgevingsconventie. Als er slechts één scène bestaat, krijgt de nieuwe scène de naam "Scene 2".

```
fl.getDocumentDOM().addNewScene();
```

## document.addNewText()

### Beschikbaarheid

Flash MX 2004; optionele parameter *text* toegevoegd in Flash CS3 Professional.

### Gebruik

```
document.addNewText(boundingBox [, text ])
```

### Parameters

**boundingRectangle** Hiermee worden de grootte en de locatie van het tekstveld aangegeven. Meer informatie over het formaat van *boundingRectangle* vindt u in [document.addNewRectangle\(\)](#).

**text** Een optionele tekenreeks met de tekst die in het veld wordt geplaatst. Als u deze parameter weglaat, wordt het gereedschap Tekst geselecteerd in het deelvenster Gereedschappen. Als u niet wilt dat een ander gereedschap wordt geselecteerd, moet u een waarde opgeven voor *text*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuw tekstveld ingevoegd en eventueel tekst in het veld geplaatst. Als u de parameter *text* weglaat, kunt u [document.setTextString\(\)](#) aanroepen om het tekstveld te vullen.

### Voorbeeld

In het volgende voorbeeld wordt een nieuw tekstveld gemaakt in de linkerbovenhoek van het werkgebied en wordt de tekst ingesteld op "Hello World":

```
fl.getDocumentDOM().addNewText({left:0, top:0, right:100, bottom:100} , "Hello World!" );  
fl.getDocumentDOM().setTextString('Hello World!');
```

### Zie ook

[document.setTextString\(\)](#)

## document.align()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.align(alignmode [, bUseDocumentBounds])
```

### Parameters

**alignmode** Een tekenreeks die aangeeft hoe de selectie moet worden uitgelijnd. Acceptabele waarden zijn "left", "right", "top", "bottom", "vertical center" en "horizontal center".

**bUseDocumentBounds** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt uitgelijnd op de grenzen van het document. Anders worden de grenzen van de geselecteerde objecten gebruikt. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.



### Beschrijving

Methode; hiermee wordt de selectie uitgelijnd.

### Voorbeeld

In het volgende voorbeeld worden objecten links en op het werkgebied uitgelijnd. Dit is hetzelfde als het inschakelen van de instelling Met werkgebied in het deelvenster Uitlejnen en klikken op de knop Links uitlijnen:

```
fl.getDocumentDOM().align("left", true);
```

### Zie ook

[document.distribute\(\)](#), [document.getAlignToDocument\(\)](#), [document.setAlignToDocument\(\)](#)

## document.allowScreens()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.allowScreens()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als `document.screenOutline` veilig kan worden gebruikt, anders `false`.

### Beschrijving

Methode; gebruik deze voordat u de eigenschap `document.screenOutline` gebruikt. Als deze methode de waarde `true` retourneert, kunt u `document.screenOutline` veilig benaderen; er wordt een fout weergegeven als u `document.screenOutline` benadert in een document zonder schermen.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of de methoden `screens` kunnen worden gebruikt in het huidige document:

```
if(fl.getDocumentDOM().allowScreens()) {  
    fl.trace("screen outline is available.");  
}  
else {  
    fl.trace("whoops, no screens.");  
}
```

### Zie ook

[document.screenOutline](#)

## document.arrange()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.arrange( arrangeMode )
```

### Parameters

**arrangeMode** Hiermee geeft u de richting op voor het verplaatsen van de selectie. Acceptabele waarden zijn "back", "backward", "forward" en "front". U hebt hiermee dezelfde mogelijkheden als met deze opties in het menu Wijzigen > Rangschikken.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie in het werkgebied gerangschikt. Deze methode is alleen van toepassing op objecten die geen vorm zijn.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie naar de voorgrond verplaatst:

```
fl.getDocumentDOM().arrange("front");
```

## document.as3AutoDeclare

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.as3AutoDeclare
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de instanties die in het werkgebied worden geplaatst, automatisch worden toegevoegd aan tijdelijklassen die door de gebruiker zijn gedefinieerd. De standaardwaarde is `true`.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat instanties die in het werkgebied in het huidige document worden geplaatst, handmatig moeten worden toegevoegd aan tijdelijklassen die door de gebruiker zijn gedefinieerd.

```
fl.getDocumentDOM().as3AutoDeclare=false;
```

## document.as3Dialect

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.as3Dialect
```

### Beschrijving

Eigenschap; een tekenreeks die het ActionScript 3.0-dialect beschrijft dat in het opgegeven document wordt gebruikt. De standaardwaarde is "AS3". Als u prototypeklassen wilt toestaan, zoals is toegestaan in oudere ECMAScript-specificaties, stelt u deze waarde in op "ES".

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat in het huidige document het dialect ECMAScript wordt gebruikt:

```
fl.getDocumentDOM().as3Dialect="ES";
```

### Zie ook

[document.asVersion](#)

## document.as3ExportFrame

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.as3ExportFrame
```

### Beschrijving

Eigenschap; een geheel getal dat aangeeft naar welk frame ActionScript 3.0-klassen worden geëxporteerd. Klassen worden standaard naar Frame 1 geëxporteerd.

### Voorbeeld

In het volgende voorbeeld wordt het frame waarnaar klassen worden geëxporteerd, gewijzigd van 1 (de standaardwaarde) in 5.

```
var myDocument = fl.getDocumentDOM();  
fl.outputPanel.trace("'Export classes in frame:' value before modification is " +  
myDocument.as3ExportFrame);  
myDocument.as3ExportFrame = 5;  
fl.outputPanel.trace("'Export classes in frame:' value after modification is " +  
myDocument.as3ExportFrame);
```

## document.as3StrictMode

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.as3StrictMode
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de ActionScript 3.0-compiler moet compileren terwijl de strikte modus is ingeschakeld (`true`) of uitgeschakeld (`false`). In de strikte modus worden waarschuwingen als fouten gerapporteerd. Dat betekent dat het compileren niet wordt voltooid als er fouten zijn. De standaardwaarde is `true`.

### Voorbeeld

In het volgende voorbeeld wordt de strikte compilermodus uitgeschakeld.

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Strict Mode value before modification is " + myDocument.as3StrictMode);
myDocument.as3StrictMode = false;
fl.outputPanel.trace("Strict Mode value after modification is " + myDocument.as3StrictMode);
```

### Zie ook

[document.as3WarningsMode](#)

## document.as3WarningsMode

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.as3WarningsMode
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de ActionScript 3.0-compiler moet compileren terwijl de waarschuwingsmodus is ingeschakeld (`true`) of uitgeschakeld (`false`). In de waarschuwingsmodus worden extra waarschuwingen gerapporteerd die u kunt gebruiken om incompatibiliteiten vast te stellen bij het bijwerken van ActionScript 2.0-code naar ActionScript 3.0. De standaardwaarde is `true`.

### Voorbeeld

In het volgende voorbeeld wordt de waarschuwingsmodus voor compileren uitgeschakeld.

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Warnings Mode value before modification is " +
myDocument.as3WarningsMode);
myDocument.as3WarningsMode = false;
fl.outputPanel.trace("Warnings Mode value after modification is " +
myDocument.as3WarningsMode);
```

### Zie ook

[document.as3StrictMode](#)

## document.asVersion

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`document.asVersion`

### Beschrijving

Eigenschap; een geheel getal dat aangeeft welke versie van ActionScript in het opgegeven bestand wordt gebruikt. Acceptabele waarden zijn 1, 2 en 3.

Als u de bedoelde playerversie voor het opgegeven document wilt bepalen, gebruikt u [document.getPlayerVersion\(\)](#). Met deze methode wordt een tekenreeks geretourneerd, zodat deze door Flash® Lite™ players kan worden gebruikt.

### Voorbeeld

In het volgende voorbeeld wordt de versie van ActionScript in het huidige document op ActionScript 2.0 ingesteld als momenteel ActionScript 1.0 is ingesteld.

```
if(fl.getDocumentDOM().asVersion == 1){  
    fl.getDocumentDOM().asVersion = 2;  
}
```

### Zie ook

[document.as3Dialect](#), [document.getPlayerVersion\(\)](#)

## document.autoLabel

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.autoLabel`

### Beschrijving

Eigenschap; een Booleaanse waarde die gelijk is aan het selectievakje Automatisch label in het deelvenster Toegankelijkheid. Met deze eigenschap kunt u aangeven dat objecten in het werkgebied automatisch worden gelabeld met de bijbehorende tekst.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `autoLabel` opgehaald en wordt het resultaat weergegeven in het deelvenster Uitvoer:

```
var isAutoLabel = fl.getDocumentDOM().autoLabel;  
fl.trace(isAutoLabel);
```

In het volgende voorbeeld wordt de eigenschap `autoLabel` ingesteld op `true`, zodat objecten in het werkgebied automatisch worden gelabeld:

```
fl.getDocumentDOM().autoLabel = true;
```

## document.backgroundColor

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.backgroundColor
```

### Beschrijving

Eigenschap; de kleur van de achtergrond, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

### Voorbeeld

In het volgende voorbeeld wordt de achtergrondkleur op zwart ingesteld:

```
fl.getDocumentDOM().backgroundColor = '#000000';
```

## document.breakApart()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.breakApart()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie gesplitst.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie gesplitst:

```
fl.getDocumentDOM().breakApart();
```

## document.canEditSymbol()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.canEditSymbol()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het menu Symbolen bewerken en de functionaliteit kunnen worden gebruikt, anders `false`.

### Beschrijving

Methode; hiermee wordt aangegeven of het menu Symbolen bewerken en de functionaliteit zijn ingeschakeld. Dit staat los van het feit of de selectie kan worden bewerkt. Deze methode moet niet worden gebruikt om te testen of

`fl.getDocumentDOM().enterEditMode()` is toegestaan.

### Voorbeeld

In het volgende voorbeeld worden de status van het menu Symbolen bewerken en de functionaliteit weergegeven in het deelvenster Uitvoer:

```
fl.trace("fl.getDocumentDOM().canEditSymbol() returns: " +  
fl.getDocumentDOM().canEditSymbol());
```

## document.canRevert()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.canRevert()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als u de methode `document.revert()` of `fl.revertDocument()` kunt gebruiken; anders `false`.

### Beschrijving

Methode; hiermee wordt bepaald of u de methode `document.revert()` of `fl.revertDocument()` kunt gebruiken.

### Voorbeeld

In het volgende voorbeeld wordt gecontroleerd of de laatst opgeslagen versie van het huidige document kan worden hersteld. Als dat het geval is, wordt de laatst opgeslagen versie hersteld met `fl.getDocumentDOM().revert()`.

```
if (fl.getDocumentDOM().canRevert()) {  
    fl.getDocumentDOM().revert();  
}
```

## document.canTestMovie()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.canTestMovie()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als u de methode `document.testMovie()` kunt gebruiken; anders `false`.

### Beschrijving

Methode; hiermee wordt bepaald of u de methode `document.testMovie()` kunt gebruiken.

### Voorbeeld

In het volgende voorbeeld wordt getest of `fl.getDocumentDOM().testMovie()` kan worden gebruikt. Als dat het geval is, wordt de methode aangeroepen.

```
if (fl.getDocumentDOM().canTestMovie()) {  
    fl.getDocumentDOM().testMovie();  
}
```

### Zie ook

[document.canTestScene\(\)](#), [document.testScene\(\)](#)

## document.canTestScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.canTestScene()
```



### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als u de methode `document.testScene()` kunt gebruiken, anders `false`.

### Beschrijving

Methode; hiermee wordt bepaald of u de methode `document.testScene()` kunt gebruiken.

### Voorbeeld

In het volgende voorbeeld wordt getest of `fl.getDocumentDOM().testScene()` kan worden gebruikt. Als dat het geval is, wordt de methode aangeroepen.

```
if (fl.getDocumentDOM().canTestScene()) {  
    fl.getDocumentDOM().testScene();  
}
```

### Zie ook

[document.canTestMovie\(\)](#), [document.testMovie\(\)](#)

## document.changeFilterOrder()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.changeFilterOrder(oldIndex, newIndex)
```

### Parameters

**oldIndex** Een geheel getal dat de huidige, op nul gebaseerde indexpositie aangeeft van het filter dat u in de lijst met filters wilt verplaatsen.

**newIndex** Een geheel getal dat de nieuwe indexpositie van het filter in de lijst aangeeft.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de index van het filter in de lijst met filters gewijzigd. Alle filters boven of onder *newIndex* worden naar boven of naar beneden verplaatst. Als u bijvoorbeeld de opdracht `fl.getDocumentDOM().changeFilterOrder(3, 0)` uitvoert met de filters die hieronder worden weergegeven, worden de filters als volgt gerangschikt:

Voor	Na
<code>blurFilterdropShadowFilterglowFiltergradientBevelFilter</code>	<code>gradientBevelFilterblurFilterdropShadowFilterglowFilter</code>

Als u vervolgens de opdracht `fl.getDocumentDOM().changeFilterOrder (0, 2)` uitvoert, worden de filters als volgt gerangschikt:

Voor	Na
<code>gradientBevelFilterblurFilterdropShadowFilterglowFilter</code>	<code>blurFilterdropShadowFiltergradientBevelFilterglowFilter</code>

### Voorbeeld

In het volgende voorbeeld wordt het filter dat op de tweede positie in de lijst met filters staat, naar de eerste positie verplaatst:

```
fl.getDocumentDOM().changeFilterOrder(1,0);
```

### Zie ook

[document.addFilter\(\)](#), [document.disableFilter\(\)](#), [document.enableFilter\(\)](#), [document.getFilters\(\)](#), [document.removeFilter\(\)](#), [Object Filter](#)

## document.clipCopy()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.clipCopy()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie vanuit het document naar het klembord gekopieerd.

Als u een tekenreeks naar het klembord wilt kopiëren, gebruikt u [fl.clipCopyString\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie vanuit het document naar het klembord gekopieerd.

```
fl.getDocumentDOM().clipCopy();
```

### Zie ook

[document.clipCut\(\)](#), [document.clipPaste\(\)](#)

## document.clipCut()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.clipCut()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie uit het document geknipt en naar het klembord geschreven.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie uit het document geknipt en naar het klembord geschreven:

```
fl.getDocumentDOM().clipCut();
```

### Zie ook

[document.clipCopy\(\)](#), [document.clipPaste\(\)](#), [fl.clipCopyString\(\)](#)

## document.clipPaste()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.clipPaste([bInPlace])
```

### Parameters

**bInPlace** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt een plakbewerking op dezelfde plaats uitgevoerd. De standaardwaarde is `false` waarbij de plakbewerking in het midden van het document wordt uitgevoerd. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de inhoud van het klembord in het document geplakt.

### Voorbeeld

In het volgende voorbeeld wordt de inhoud van het klembord in het midden van het document geplakt:

```
fl.getDocumentDOM().clipPaste();
```

In het volgende voorbeeld wordt de inhoud van het klembord op dezelfde plaats in het document geplakt:

```
fl.getDocumentDOM().clipPaste(true);
```

#### Zie ook

[document.clipCopy\(\)](#), [document.clipCut\(\)](#), [fl.clipCopyString\(\)](#)

## document.close()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.close([bPromptToSaveChanges])
```

### Parameters

**bPromptToSaveChanges** Een Booleaanse waarde: als de waarde is ingesteld op `true`, verschijnt er een dialoogvenster voor de gebruiker als het document niet-opgeslagen wijzigingen bevat. Als *bPromptToSaveChanges* is ingesteld op `false`, wordt niet aan de gebruiker gevraagd of gewijzigde documenten moeten worden opgeslagen. De standaardwaarde is `true`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het opgegeven document gesloten.

### Voorbeeld

In het volgende voorbeeld wordt het huidige document gesloten en wordt een dialoogvenster voor het opslaan van wijzigingen weergegeven:

```
fl.getDocumentDOM().close();
```

In het volgende voorbeeld wordt het huidige document gesloten zonder wijzigingen op te slaan:

```
fl.getDocumentDOM().close(false);
```

## document.convertLinesToFills()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.convertLinesToFills()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden lijnen geconverteerd naar vullingen voor de geselecteerde objecten.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde lijnen geconverteerd naar vullingen:

```
fl.getDocumentDOM().convertLinesToFills();
```

## document.convertToSymbol()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.convertToSymbol(type, name, registrationPoint)
```

### Parameters

**type** Een tekenreeks die het type aangeeft van het symbool dat moeten worden gemaakt. Acceptabele waarden zijn "movie clip", "button" en "graphic".

**name** Een tekenreeks met de naam voor het nieuwe symbool; deze moet uniek zijn. Als u een lege tekenreeks opgeeft, maakt de methode een unieke symboolnaam voor u.

**registratiepunt** Hiermee wordt het punt aangegeven dat de locatie 0,0 voor het symbool bepaalt. Acceptabele waarden zijn: "top left", "top center", "top right", "center left", "center", "center right", "bottom left", "bottom center" en "bottom right".

### Geretourneerde waarde

Een object voor het nieuwe symbool of null als het symbool niet kan worden gemaakt.

### Beschrijving

Methode; hiermee worden een of meer geselecteerde items in het werkgebied geconverteerd naar een nieuw symbool. Zie [Object Item](#) voor informatie over het definiëren van koppelingen en eigenschappen van gedeelde elementen voor een symbool.

### Voorbeeld

In de volgende voorbeelden worden een filmclipsymbool met een opgegeven naam, een knopsymbool met een opgegeven naam en een filmclipsymbool met een standaardnaam gemaakt:

```
newMc = fl.getDocumentDOM().convertToSymbol("movie clip", "mcSymbolName", "top left");  
newButton = fl.getDocumentDOM().convertToSymbol("button", "btnSymbolName", "bottom right");  
newClipWithDefaultName = fl.getDocumentDOM().convertToSymbol("movie clip", "", "top left");
```

## document.crop()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.crop()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee wordt het bovenste geselecteerde tekenobject gebruikt om alle geselecteerde tekenobjecten eronder bij te snijden. Deze methode retourneert `false` als er geen tekenobjecten zijn geselecteerd of als een of meer geselecteerde items geen tekenobjecten zijn.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde objecten bijgesneden:

```
fl.getDocumentDOM().crop();
```

### Zie ook

[document.deleteEnvelope\(\)](#), [document.intersect\(\)](#), [document.punch\(\)](#), [document.union\(\)](#),  
[shape.isDrawingObject](#)

## document.currentPublishProfile

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.currentPublishProfile
```

### Beschrijving

Eigenschap; een tekenreeks met de naam van het actieve publicatieprofiel voor het opgegeven document.

### Voorbeeld

In het volgende voorbeeld wordt een nieuw publicatieprofiel met de standaardnaam toegevoegd en wordt de naam van het profiel weergegeven in het deelvenster Uitvoer:

```
fl.getDocumentDOM().addNewPublishProfile();  
fl.outputPanel.trace(fl.getDocumentDOM().currentPublishProfile);
```

In het volgende voorbeeld wordt het geselecteerde publicatieprofiel gewijzigd in "Default":

```
fl.getDocumentDOM().currentPublishProfile = "Default";
```

## document.currentTimeline

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.currentTimeline
```

### Beschrijving

Eigenschap; een geheel getal dat de index van de actieve tijdlijn aangeeft. U kunt de actieve tijdlijn instellen door de waarde van deze eigenschap te wijzigen; het effect is bijna gelijk aan het aanroepen van `document.editScene()`. Het enige verschil is dat er geen foutbericht wordt weergegeven als de index van de tijdlijn niet geldig is. De eigenschap wordt niet ingesteld, waardoor een stille fout optreedt.

### Voorbeeld

In het volgende voorbeeld wordt de index van de huidige tijdlijn weergegeven:

```
var myCurrentTL = fl.getDocumentDOM().currentTimeline;  
fl.trace("The index of the current timeline is: "+ myCurrentTL);
```

In het volgende voorbeeld wordt de actieve tijdlijn gewijzigd in een scène met de naam "myScene":

```
var i = 0;  
var curTimelines = fl.getDocumentDOM().timelines;  
while(i < fl.getDocumentDOM().timelines.length){  
    if(curTimelines[i].name == "myScene"){  
        fl.getDocumentDOM().currentTimeline = i;  
    }  
    ++i;  
}
```

### Zie ook

[document.getTimeline\(\)](#)

## document.debugMovie()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
document.DebugMovie([Boolean abortIfErrorsExist])
```

### Beschrijving

Methode, hiermee wordt de opdracht voor het opsporen van fouten in de film aangeroepen voor het document.

### Parameters

**abortIfErrorsExist** Booleaanse waarde, de standaardinstelling is false. Als deze wordt ingesteld op true, wordt de foutopsporingsessie niet gestart en wordt het SWF-venster niet geopend in geval van compilatiefouten. Compilatiewaarschuwingen leiden niet tot het afbreken van de opdracht.

### Voorbeeld

In het volgende voorbeeld wordt het huidige document geopend in de foutopsporingsmodus, maar wordt de bewerking afgebroken in geval van compilatiefouten:

```
fl.getDocumentDOM().debugMovie(1);
```

## document.deleteEnvelope()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.deleteEnvelope()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee wordt de envelop (kader dat een of meer objecten bevat) van de geselecteerde objecten verwijderd.

### Voorbeeld

In het volgende voorbeeld wordt de envelop van de geselecteerde objecten verwijderd:

```
fl.getDocumentDOM().deleteEnvelope();
```

### Zie ook

[document.crop\(\)](#), [document.intersect\(\)](#), [document.punch\(\)](#), [document.union\(\)](#), [shape.isDrawingObject](#)

## document.deletePublishProfile()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.deletePublishProfile()
```



### Parameters

Geen.

### Geretourneerde waarde

Een geheel getal dat de index van het nieuwe huidige profiel is. Als er geen nieuw profiel beschikbaar is, blijft het huidige profiel ongewijzigd en wordt index daarvan geretourneerd.

### Beschrijving

Methode; hiermee wordt het huidige actieve profiel verwijderd als er meerdere zijn. Er moet ten minste één profiel overblijven.

### Voorbeeld

In het volgende voorbeeld wordt het huidige actieve profiel verwijderd, als er meerdere zijn, en wordt de index van het nieuwe actieve profiel weergegeven:

```
alert(fl.getDocumentDOM().deletePublishProfile());
```

### Zie ook

[document.addNewPublishProfile\(\)](#)

## document.deleteScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.deleteScene()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de scène is verwijderd, anders `false`.

### Beschrijving

Methode; hiermee wordt de huidige scène ([Object Timeline](#)) verwijderd en wordt de volgende scène ingesteld als het huidige object Timeline, als de verwijderde scène niet de laatste scène was. Als de verwijderde scène de laatste scène was, wordt het eerste object ingesteld als het huidige object Timeline. Als er slechts één object Timeline (scène) bestaat, retourneert de methode de waarde `false`.

### Voorbeeld

Ervan uitgaande dat het huidige document drie scènes bevat (`Scene0`, `Scene1` en `Scene2`), wordt in het volgende voorbeeld `Scene2` de huidige scène en wordt deze daarna verwijderd:

```
fl.getDocumentDOM().editScene(2);  
var success = fl.getDocumentDOM().deleteScene();
```

## document.deleteSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.deleteSelection()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie verwijderd uit het werkgebied. Er wordt een foutbericht weergegeven als er niets is geselecteerd.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie uit het document verwijderd:

```
fl.getDocumentDOM().deleteSelection();
```

## document.description

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.description
```

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid. De beschrijving wordt gelezen door de schermlezer.

### Voorbeeld

In het volgende voorbeeld wordt de beschrijving van het document ingesteld:

```
fl.getDocumentDOM().description= "This is the main movie";
```

in het volgende voorbeeld wordt de beschrijving van het document opgehaald en weergegeven in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().description);
```

## document.disableAllFilters()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.disableAllFilters()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle filters voor de geselecteerde objecten uitgeschakeld.

### Voorbeeld

in het volgende voorbeeld worden alle filters voor de geselecteerde objecten uitgeschakeld.

```
fl.getDocumentDOM().disableAllFilters();
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#),  
[document.disableOtherFilters\(\)](#), [document.enableAllFilters\(\)](#), [document.getFilters\(\)](#),  
[document.removeAllFilters\(\)](#), [Object Filter](#)

## document.disableFilter()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.disableFilter(filterIndex)
```

### Parameters

**filterIndex** Een geheel getal dat staat voor de op nul gebaseerde index van het filter in de lijst met filters.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het opgegeven filter in de lijst met filters uitgeschakeld.

### Voorbeeld

In het volgende voorbeeld worden het eerste en het derde filter (indexwaarden 0 en 2) in de lijst met filters uitgeschakeld voor de geselecteerde objecten:

```
fl.getDocumentDOM().disableFilter(0);  
fl.getDocumentDOM().disableFilter(2);
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),  
[document.disableOtherFilters\(\)](#), [document.enableFilter\(\)](#), [document.getFilters\(\)](#),  
[document.removeFilter\(\)](#), [Object Filter](#)

## document.disableOtherFilters()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.disableOtherFilters(enabledFilterIndex)
```

### Parameters

**enabledFilterIndex** Een geheel getal dat de op nul gebaseerde index aangeeft van het filter dat ingeschakeld moet blijven nadat andere filters zijn uitgeschakeld.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle filters uitgeschakeld behalve het filter op de opgegeven positie in de lijst met filters.

### Voorbeeld

In het volgende voorbeeld worden alle filters uitgeschakeld behalve het tweede filter in de lijst (indexwaarde 1):

```
fl.getDocumentDom().disableOtherFilters(1);
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),  
[document.disableFilter\(\)](#), [document.enableFilter\(\)](#), [document.getFilters\(\)](#),  
[document.removeFilter\(\)](#), [Object Filter](#)

## document.distribute()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.distribute(distributemode [, bUseDocumentBounds])
```

### Parameters

**distributemode** Een tekenreeks die aangeeft waarover de geselecteerde objecten moeten worden verdeeld. Acceptabele waarden zijn "left edge", "horizontal center", "right edge", "top edge", "vertical center" en "bottom edge".

**bUseDocumentBounds** Een Booleaanse waarde; als de waarde is ingesteld op `true`, worden de grenzen van het document gebruikt om de geselecteerde objecten te verdelen. Anders worden de grenzen van de geselecteerde objecten gebruikt. De standaardwaarde is `false`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie verdeeld.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde objecten verdeeld op basis van de bovenrand:

```
fl.getDocumentDOM().distribute("top edge");
```

In het volgende voorbeeld worden de geselecteerde objecten verdeeld op basis van de bovenrand en wordt de parameter *bUseDocumentBounds* expliciet ingesteld:

```
fl.getDocumentDOM().distribute("top edge", false);
```

In het volgende voorbeeld worden de geselecteerde objecten verdeeld over de bovenrand, waarbij de grenzen van het document worden gebruikt:

```
fl.getDocumentDOM().distribute("top edge", true);
```

### Zie ook

[document.getAlignToDocument\(\)](#), [document.setAlignToDocument\(\)](#)

## document.distributeToLayers()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.distributeToLayers()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie verdeeld over lagen; dit is gelijk aan het selecteren van Over lagen verdelen. Er wordt een fout weergegeven als er niets is geselecteerd.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie over lagen verdeeld:

```
fl.getDocumentDOM().distributeToLayers();
```

## document.docClass

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.docClass
```

### Beschrijving

Eigenschap; een tekenreeks die de ActionScript 3.0-klasse op het hoogste niveau aangeeft die aan het document is gekoppeld. Als het document niet is geconfigureerd voor het gebruik van ActionScript 3.0, wordt deze eigenschap genegeerd.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat de ActionScript 3.0-klasse die aan het document is gekoppeld, `com.mycompany.ManagerClass` is. Dit is in `com/mycompany/ManagerClass` gedefinieerd als:

```
var myDocument = fl.getDocumentDOM();  
// set the property  
myDocument.docClass = "com.mycompany.ManagerClass";  
// get the property  
fl.outputPanel.trace("document.docClass has been set to " + myDocument.docClass);
```

### Zie ook

[item.linkageBaseClass](#)

## document.documentHasData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.documentHasData(name)
```

### Parameters

**name** Een tekenreeks waarmee de naam wordt opgegeven van de gegevens die moeten worden gecontroleerd.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het document permanente gegevens heeft, anders `false`.

### Beschrijving

Methode; hiermee wordt gecontroleerd of het document permanente gegevens met de opgegeven naam bevat.

### Voorbeeld

In het volgende voorbeeld wordt gecontroleerd of het document permanente gegevens met de naam "myData" bevat:

```
var hasData = fl.getDocumentDOM().documentHasData("myData");
```

### Zie ook

[document.addDataToDocument\(\)](#), [document.getDataFromDocument\(\)](#),  
[document.removeDataFromDocument\(\)](#)

## document.duplicatePublishProfile()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.duplicatePublishProfile([profileName])
```

### Parameters

**profileName** Een tekenreeks waarmee de unieke naam van het gedupliceerde profiel wordt opgegeven. Als u geen naam opgeeft, wordt de standaardnaam gebruikt. Deze parameter is optioneel.

### Geretourneerde waarde

Een geheel getal dat de index is van het nieuwe profiel in de lijst met profielen. Retourneert -1 als het profiel niet kan worden gedupliceerd.

### Beschrijving

Methode; hiermee wordt het huidige actieve profiel gedupliceerd en krijgt de gedupliceerde versie de focus.

### Voorbeeld

In het volgende voorbeeld wordt het huidige actieve profiel gedupliceerd en wordt de index van het nieuwe profiel weergegeven in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().duplicatePublishProfile("dup profile"));
```

## document.duplicateScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.duplicateScene()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de scène is gedupliceerd, anders `false`.

### Beschrijving

Methode; hiermee wordt een kopie gemaakt van de geselecteerde scène, de nieuwe scène een unieke naam gegeven en ingesteld als de huidige scène.

### Voorbeeld

In het volgende voorbeeld wordt de tweede scène in het huidige document gedupliceerd:

```
fl.getDocumentDOM().editScene(1); //Set the middle scene to current scene.  
var success = fl.getDocumentDOM().duplicateScene();
```

## document.duplicateSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.duplicateSelection()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie in het werkgebied gedupliceerd.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie gedupliceerd; dit is hetzelfde als de Alt-toets ingedrukt houden terwijl u een item sleept:

```
fl.getDocumentDOM().duplicateSelection();
```

## document.editScene()

### Beschikbaarheid

Flash MX 2004.



### Gebruik

```
document.editScene(index)
```

### Parameters

**index** Een op nul gebaseerd geheel getal dat aangeeft welke scène moet worden bewerkt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de opgegeven scène geselecteerd voor bewerken.

### Voorbeeld

Ervan uitgaande dat het huidige document drie scènes bevat (*Scene0*, *Scene1* en *Scene2*), wordt in het volgende voorbeeld *Scene2* de huidige scène en wordt deze daarna verwijderd:

```
fl.getDocumentDOM().editScene(2);  
fl.getDocumentDOM().deleteScene();
```

## document.enableAllFilters()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.enableAllFilters()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle filters in de lijst met filters ingeschakeld voor een of meer geselecteerde objecten.

### Voorbeeld

In het volgende voorbeeld worden alle filters in de lijst met filters ingeschakeld voor een of meer geselecteerde objecten:

```
fl.getDocumentDOM().enableAllFilters();
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#),  
[document.enableFilter\(\)](#), [document.getFilters\(\)](#), [document.removeAllFilters\(\)](#), [Object Filter](#)

## document.enableFilter()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.enableFilter(filterIndex)
```

### Parameters

**filterIndex** Een geheel getal dat de op nul gebaseerde index aangeeft van het filter in de lijst met filters dat moet worden ingeschakeld.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het opgegeven filter ingeschakeld voor een of meer geselecteerde objecten.

### Voorbeeld

In het volgende voorbeeld wordt het tweede filter ingeschakeld voor een of meer geselecteerde objecten:

```
fl.getDocumentDOM().enableFilter(1);
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#),  
[document.enableAllFilters\(\)](#), [document.getFilters\(\)](#), [document.removeFilter\(\)](#), [Object Filter](#)

## document.enterEditMode()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.enterEditMode([editMode])
```

### Parameters

**editMode** Een tekenreeks waarmee de bewerkingsmodus wordt opgegeven. Acceptabele waarden zijn "inPlace" of "newWindow". Als u geen parameter opgeeft, wordt de symboolbewerkingsmodus gactiveerd. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt geschakeld naar de bewerkingsmodus die is aangegeven met de parameter. Als er geen parameter is opgegeven, wordt de symboolbewerkingsmodus geactiveerd; dit heeft hetzelfde resultaat als met de rechtermuisknop op het symbool klikken om het contextmenu te openen en vervolgens Bewerken selecteren.

### Voorbeeld

In het volgende voorbeeld wordt de modus Op plaats bewerken geactiveerd voor het geselecteerde symbool:

```
fl.getDocumentDOM().enterEditMode('inPlace');
```

In het volgende voorbeeld wordt de bewerkingsmodus in een nieuw venster geactiveerd voor het geselecteerde symbool:

```
fl.getDocumentDOM().enterEditMode('newWindow');
```

### Zie ook

[document.exitEditMode\(\)](#)

## document.exitEditMode()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.exitEditMode()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de symboolbewerkingsmodus afgesloten en wordt het niveau boven de bewerkingsmodus geactiveerd. Als u bijvoorbeeld een symbool in een ander symbool bewerkt, gaat u met deze methode van het symbool dat u bewerkt, naar het bovenliggende symbool.

### Voorbeeld

In het volgende voorbeeld wordt de symboolbewerkingsmodus afgesloten:

```
fl.getDocumentDOM().exitEditMode();
```

### Zie ook

[document.enterEditMode\(\)](#)

## document.exportPNG()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.exportPNG([fileURI [, bCurrentPNGSettings [, bCurrentFrame]])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de bestandsnaam van het geëxporteerde bestand aangeeft. Als *fileURI* een lege tekenreeks is of niet is opgegeven, wordt het dialoogvenster Film exporteren weergegeven.

**bCurrentPNGSettings** Een Booleaanse waarde die aangeeft of de huidige PNG-publicatie-instellingen worden gebruikt (*true*) of dat het dialoogvenster PNG-bestand exporteren wordt weergegeven (*false*). Deze parameter is optioneel. De standaardwaarde is *false*.

**bCurrentFrame** Een Booleaanse waarde die aangeeft of alleen het huidige frame wordt geëxporteerd (*true*) of dat alle frames worden geëxporteerd, met voor elk frame een apart PNG-bestand (*false*). Deze parameter is optioneel. De standaardwaarde is *false*.

### Geretourneerde waarde

De Booleaanse waarde *true* als het bestand is geëxporteerd als PNG-bestand, anders *false*.

### Beschrijving

Methode; hiermee wordt het document geëxporteerd als een of meer PNG-bestanden. Als *fileURI* is opgegeven en het bestand al bestaat, wordt het zonder waarschuwing overschreven.

### Voorbeeld

In het volgende voorbeeld wordt het huidige frame in het huidige document geëxporteerd naar *myFile.png* waarbij de huidige PNG-publicatie-instellingen worden gebruikt:

```
fl.getDocumentDOM().exportPNG("file:///C:/myProject/myFile.png", true, true);
```

## document.exportPublishProfile()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.exportPublishProfile(fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad aangeeft van het XML-bestand waarnaar het profiel wordt geëxporteerd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het huidige actieve profiel geëxporteerd naar een XML-bestand.

### Voorbeeld

In het volgende voorbeeld wordt het huidige actieve profiel geëxporteerd naar het bestand Profile.xml in de map /Documents and Settings/gebruikersnaam/Bureaublad op station C:

```
fl.getDocumentDOM().exportPublishProfile('file:///C:/Documents and  
Settings/username/Desktop/profile.xml');
```

### Zie ook

[document.exportPublishProfileString\(\)](#), [document.importPublishProfile\(\)](#)

## document.exportPublishProfileString()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.exportPublishProfileString( [profileName] )
```

### Parameters

**profileName** Een tekenreeks met de naam van het profiel dat moet worden geëxporteerd naar een XML-tekenreeks. Deze parameter is optioneel.

### Geretourneerde waarde

Een XML-tekenreeks.

### Beschrijving

Methode; hiermee wordt een tekenreeks geretourneerd met het opgegeven profiel in XML-indeling. Als u geen waarde voor *profileName* opgeeft, wordt het huidige profiel geëxporteerd.

### Voorbeeld

In het volgende voorbeeld wordt een XML-tekenreeks met het huidige profiel opgeslagen in een variabele met de naam *profileXML* en vervolgens weergegeven in het deelvenster Uitvoer:

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString();  
fl.trace(profileXML);
```

### Zie ook

[document.exportPublishProfile\(\)](#), [document.importPublishProfileString\(\)](#)

## document.exportSWF()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.exportSWF([fileURI [, bCurrentSettings]])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de naam van het geëxporteerde bestand aangeeft. Als *fileURI* leeg is of niet is opgegeven, wordt het dialoogvenster Film exporteren weergegeven. Deze parameter is optioneel.

**bCurrentSettings** Een Booleaanse waarde; als de waarde is ingesteld op `true`, worden de huidige SWF-publicatie-instellingen gebruikt. Anders wordt het dialoogvenster Flash Player exporteren weergegeven. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het document geëxporteerd in de Flash SWF-indeling.

### Voorbeeld

In het volgende voorbeeld wordt het document geëxporteerd naar de opgegeven bestandslocatie met de huidige publicatie-instellingen:

```
fl.getDocumentDOM().exportSWF("file:///C:/Documents and  
Settings/joe_user/Desktop/qwerty.swf");
```

In het volgende voorbeeld worden het dialoogvenster Film exporteren en het dialoogvenster Flash Player exporteren weergegeven en wordt het document vervolgens geëxporteerd op basis van de opgegeven instellingen:

```
fl.getDocumentDOM().exportSWF("", true);
```

In het volgende voorbeeld wordt het dialoogvenster Film exporteren weergegeven en wordt het document vervolgens geëxporteerd op basis van de opgegeven instellingen:

```
fl.getDocumentDOM().exportSWF();
```

## document.externalLibraryPath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.externalLibraryPath
```

### Beschrijving

Eigenschap; een tekenreeks die een lijst bevat met items in het externe ActionScript 3.0-bibliotheekpad van het document, dat de locatie aangeeft van SWC-bestanden die worden gebruikt als gedeelde runtime bibliotheken. Items in de tekenreeks worden gescheiden door puntkomma's. In het ontwerpgereedschap worden de items opgegeven door Bestand > Publicatie-instellingen te kiezen en vervolgens ActionScript 3.0-instellingen te kiezen op het tabblad Flash.

### Voorbeeld

In het volgende voorbeeld wordt het externe bibliotheekpad van het document ingesteld op "." en "../mySWCLibrary":

```
var myDocument = fl.getDocumentDOM();  
myDocument.externalLibraryPath = ".;../mySWCLibrary";  
fl.trace(myDocument.externalLibraryPath);
```

### Zie ook

[document.libraryPath](#), [document.sourcePath](#), [fl.externalLibraryPath](#)

## document.forceSimple

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.forceSimple
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de onderliggende items van het opgegeven object toegankelijk zijn. Dit is gelijk aan de omgekeerde logica van de instelling Onderliggende objecten toegankelijk maken in het deelvenster Toegankelijkheid. Dit betekent dat `forceSimple` is `true`, hetzelfde is als het uitschakelen van de optie Onderliggend object toegankelijk maken. `forceSimple` is `false`, is hetzelfde als het inschakelen van de optie Onderliggend object toegankelijk maken.

### Voorbeeld

In het volgende voorbeeld wordt de variabele `areChildrenAccessible` ingesteld op de waarde van de eigenschap `forceSimple`. De waarde `false` betekent dat de onderliggende items toegankelijk zijn.

```
var areChildrenAccessible = fl.getDocumentDOM().forceSimple;
```

In het volgende voorbeeld wordt de eigenschap `forceSimple` zo ingesteld dat de onderliggende items van het document toegankelijk zijn:

```
fl.getDocumentDOM().forceSimple = false;
```

## document.frameRate

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.frameRate
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die aangeeft hoeveel frames per seconde worden weergegeven wanneer het SWF-bestand wordt afgespeeld; de standaardwaarde is 12. Het instellen van deze eigenschap is hetzelfde als het instellen van de standaard framesnelheid in het dialoogvenster Documenteigenschappen (Wijzigen > Document) in het FLA-bestand.

### Voorbeeld

In het volgende voorbeeld wordt de framesnelheid ingesteld op 25,5 frames per seconde:

```
fl.getDocumentDOM().frameRate = 25.5;
```

## document.getAlignToDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getAlignToDocument()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de voorkeursinstelling aangeeft dat objecten met het werkgebied worden uitgelijnd, anders `false`.

### Beschrijving

Methode; identiek aan het ophalen van de waarde van de knop Met werkgebied in het deelvenster Uitlijnen. Hiermee haalt u de voorkeuren op die kunnen worden gebruikt voor de methoden [document.align\(\)](#), [document.distribute\(\)](#), [document.match\(\)](#) en [document.space\(\)](#) voor het document.

### Voorbeeld

In het volgende voorbeeld wordt de waarde opgehaald van de knop Met werkgebied in het deelvenster Uitlijnen. Als `true` wordt geretourneerd, is de knop Met werkgebied actief, anders is de knop niet actief.

```
var isAlignToDoc = fl.getDocumentDOM().getAlignToDocument();  
fl.getDocumentDOM().align("left", isAlignToDoc);
```

### Zie ook

[document.setAlignToDocument\(\)](#)

## document.getBlendMode()

### Beschikbaarheid

Flash 8.



### Gebruik

```
document.getBlendMode()
```

### Parameters

Geen.

### Geretourneerde waarde

Een tekenreeks waarmee de overvloeimodus voor een of meer geselecteerde objecten wordt opgegeven. Als er meerdere objecten zijn geselecteerd en deze verschillende overvloeimodi hebben, heeft de tekenreeks betrekking op de overvloeimodus van het object met de grootste diepte.

**Opmerking:** De geretourneerde waarde is onvoorspelbaar als de selectie objecten bevat die geen overvloeimodi ondersteunen of waarvoor de waarde van de overvloeimodus "normal" is.

### Beschrijving

Methode; hiermee wordt een tekenreeks geretourneerd waarmee de overvloeimodus voor een of meer geselecteerde objecten wordt opgegeven.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de overvloeimodus weergegeven in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDom().getBlendMode());
```

## document.getCustomFill()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getCustomFill([objectToFill])
```

### Parameters

**objectToFill** Een tekenreeks die de locatie van het vulobject bevat. U kunt de volgende waarden gebruiken:

- "toolbar" retourneert het vulobject van het deelvenster Gereedschappen en Eigenschapcontrole.
- "selection" retourneert het vulobject van de selectie.

Als u deze parameter weglaat, is "selection" de standaardwaarde. Als er niets is geselecteerd, retourneert de methode `undefined`. Deze parameter is optioneel.

### Geretourneerde waarde

Het [Object Fill](#) dat wordt opgegeven met de parameter `objectToFill` als de bewerking succesvol was, anders wordt `undefined` geretourneerd.

### Beschrijving

Methode; hiermee wordt het vulobject opgehaald van de geselecteerde vorm of, indien dat is opgegeven, van het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In het volgende voorbeeld wordt het vulobject van de selectie opgehaald en wordt de kleur van de selectie vervolgens gewijzigd in wit:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = '#FFFFFF';  
fill.style = "solid";  
fl.getDocumentDOM().setCustomFill(fill);
```

In het volgende voorbeeld wordt het vulobject van het deelvenster Gereedschappen en Eigenschapcontrole geretourneerd en wordt de kleurstaal vervolgens gewijzigd in een lineair verloop:

```
var fill = fl.getDocumentDOM().getCustomFill("toolbar");  
fill.style = "linearGradient";  
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];  
fill.posArray = [0, 100, 200];  
fl.getDocumentDOM().setCustomFill( fill );
```

### Zie ook

[document.setCustomFill\(\)](#)

## document.getCustomStroke()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getCustomStroke([locationOfStroke])
```

### Parameters

**locationOfStroke** Een tekenreeks die de locatie van het streekobject bevat. U kunt de volgende waarden gebruiken:

- "toolbar", retourneert het streekobject van het deelvenster Gereedschappen en Eigenschapcontrole (indien dit is ingesteld).
- "selection", retourneert het streekobject van de selectie (indien dit is ingesteld).

Als u deze parameter weglaat, is "selection" de standaardwaarde. Als er niets is geselecteerd, wordt *undefined* geretourneerd. Deze parameter is optioneel.

### Geretourneerde waarde

Het [Object Stroke](#) dat wordt opgegeven met de parameter *locationOfStroke* als de bewerking succesvol was, anders wordt *undefined* geretourneerd.

### Beschrijving

Retourneert het streekobject van de geselecteerde vorm of, indien dit is opgegeven, van het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In het volgende voorbeeld worden de huidige streekinstellingen van de selectie geretourneerd en wordt de strekdikte ingesteld op 2:

```
var stroke = fl.getDocumentDOM().getCustomStroke("selection");
stroke.thickness = 2;
fl.getDocumentDOM().setCustomStroke(stroke);
```

In het volgende voorbeeld worden de huidige streekinstellingen van het deelvenster Gereedschappen en Eigenschapcontrole geretourneerd en wordt de streekkleur ingesteld op rood:

```
var stroke = fl.getDocumentDOM().getCustomStroke("toolbar");
stroke.color = "#FF0000";
fl.getDocumentDOM().setCustomStroke(stroke);
```

#### Zie ook

[document.setCustomStroke\(\)](#)

## document.getDataFromDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getDataFromDocument(name)
```

### Parameters

**name** Een tekenreeks met de naam van de gegevens die moeten worden geretourneerd.

### Geretourneerde waarde

De opgegeven gegevens.

### Beschrijving

Methode; hiermee wordt de waarde van de opgegeven gegevens opgehaald. Het type dat wordt geretourneerd, is afhankelijk van het type gegevens dat is opgeslagen.

### Voorbeeld

In het volgende voorbeeld wordt het gehele getal 12 toegevoegd aan het huidige document en wordt deze methode gebruikt om de waarde weer te geven in het deelvenster Uitvoer:

```
fl.getDocumentDOM().addDataToDocument("myData", "integer", 12);
fl.trace(fl.getDocumentDOM().getDataFromDocument("myData"));
```

#### Zie ook

[document.addDataToDocument\(\)](#), [document.documentHasData\(\)](#), [document.removeDataFromDocument\(\)](#)

## document.getElementProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getElementProperty(propertyName)
```

### Parameters

**propertyName** Een tekenreeks met de naam van de eigenschap `Element` waarvoor de waarde moet worden opgehaald.

### Geretourneerde waarde

De waarde van de opgegeven eigenschap. Retourneert `null` als de status van de eigenschap onbepaald is, bijvoorbeeld wanneer meerdere elementen met verschillende eigenschapswaarden zijn geselecteerd. Retourneert `undefined` als de eigenschap geen geldige eigenschap is van het geselecteerde element.

### Beschrijving

Methode; hiermee wordt de opgegeven eigenschap `Element` voor de huidige selectie opgehaald. Zie de overzichtstabel met eigenschappen van het [Object Element](#) voor een lijst met acceptabele waarden.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de eigenschap `Element` voor de huidige selectie opgehaald:

```
// elementName = the instance name of the selected object.  
var elementName = fl.getDocumentDOM().getElementProperty("name");
```

### Zie ook

[document.setElementProperty\(\)](#)

## document.getElementTextAttr()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getElementTextAttr(attrName [, startIndex [, endIndex]])
```

### Parameters

**attrName** Een tekenreeks met de naam van de eigenschap `TextAttrs` die moet worden geretourneerd. Zie de overzichtstabel met eigenschappen van het [Object TextAttrs](#) voor een lijst met namen van eigenschappen en verwachte waarden.

**startIndex** Een geheel getal dat de index van het eerste teken bepaalt, waarbij 0 (nul) de eerste positie aangeeft. Deze parameter is optioneel.

**endIndex** Een geheel getal dat de index van het laatste teken bepaalt. Deze parameter is optioneel.

### Geretourneerde waarde

Als er één tekstveld is geselecteerd, wordt de eigenschap geretourneerd als er slechts één waarde is gebruikt in de tekst. Retourneert `undefined` als er meerdere waarden zijn gebruikt in het tekstveld. Als er meerdere tekstvelden zijn geselecteerd en alle tekstuitlijningswaarden gelijk zijn, wordt deze waarde geretourneerd. Als er meerdere tekstvelden zijn geselecteerd, maar niet alle tekstuitlijningswaarden gelijk zijn, wordt `undefined` geretourneerd. Als de optionele argumenten niet worden doorgegeven, zijn deze regels van toepassing op de geselecteerde tekst of op het hele tekstveld als de tekst momenteel niet wordt bewerkt. Als alleen `startIndex` wordt doorgegeven, wordt de eigenschap van het teken rechts van de index geretourneerd als de waarden van alle geselecteerde tekstobjecten overeenkomen. Als `startIndex` en `endIndex` worden doorgegeven, heeft de geretourneerde waarde betrekking op het hele bereik met tekens van `startIndex` tot aan, maar niet inclusief, `endIndex`.

### Beschrijving

Methode; hiermee wordt een specifieke eigenschap `TextAttrs` van de geselecteerde tekstobjecten opgehaald. Geselecteerde objecten die geen tekstvelden zijn, worden genegeerd. Zie de overzichtstabel met eigenschappen van het [Object TextAttrs](#) voor een lijst met namen van eigenschappen en verwachte waarden. Zie ook `document.setElementTextAttr()`.

### Voorbeeld

In het volgende voorbeeld wordt de grootte van de geselecteerde tekstvelden opgehaald:

```
fl.getDocumentDOM().getElementTextAttr("size");
```

In het volgende voorbeeld wordt de kleur opgehaald van het teken op index 3 in de geselecteerde tekstvelden:

```
fl.getDocumentDOM().getElementTextAttr("fillColor", 3);
```

In het volgende voorbeeld wordt de lettertypenaam opgehaald van de tekst van index 2 tot aan, maar niet inclusief, index 10 van de geselecteerde tekstvelden:

```
fl.getDocumentDOM().getElementTextAttr("face", 2, 10);
```

## document.getFilters()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.getFilters()
```

### Parameters

Geen.

### Geretourneerde waarde

Een matrix die een lijst bevat met filters die zijn toegepast op een of meer geselecteerde objecten.

### Beschrijving

Methode; hiermee wordt een array geretourneerd die de lijst bevat met filters die zijn toegepast op een of meer geselecteerde objecten. Als er meerdere objecten zijn geselecteerd en deze niet dezelfde filters hebben, wordt een lijst geretourneerd met filters die zijn toegepast op het eerste geselecteerde object.

### Voorbeeld

Zie `document.setFilters()`.

### Zie ook

`document.addFilter()`, `document.changeFilterOrder()`, `document.setFilters()`, [Object Filter](#)

## document.getMetadata()

### Beschikbaarheid

Flash 8.

### Gebruik

`document.getMetadata()`

### Parameters

Geen.

### Geretourneerde waarde

Een tekenreeks die de XML-metagegevens bevat die aan het document zijn gekoppeld of een lege tekenreeks als er geen metagegevens zijn.

### Beschrijving

Methode; hiermee wordt een tekenreeks geretourneerd die de XML-metagegevens bevat die aan het document zijn gekoppeld of een lege tekenreeks als er geen metagegevens zijn.

### Voorbeeld

In het volgende voorbeeld worden de XML-metagegevens van het huidige document weergegeven in het deelvenster Uitvoer:

```
fl.trace("XML Metadata is :" + fl.getDocumentDOM().getMetadata());
```

### Zie ook

`document.setMetadata()`

## document.getMobileSettings()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`document.getMobileSettings()`

### Parameters

Geen.

### Geretourneerde waarde

Een tekenreeks met de XML-instellingen voor het document. Als er geen waarde is ingesteld, wordt een lege tekenreeks geretourneerd.

### Beschrijving

Methode; hiermee worden de mobiele XML-instellingen voor het document geretourneerd.

### Voorbeeld

In het volgende voorbeeld worden de XML-instellingen voor het huidige document weergegeven:

```
fl.trace(fl.getDocumentDOM().getMobileSettings());  
//traces a string like the following"<? xml version="1.0" encoding="UTF-16" standalone="no"  
?><mobileSettings> <contentType id="standalonePlayer" name="Standalone Player"/>  
<testDevices> <testDevice id="1170" name="Generic Phone" selected="yes"/> </testDevices>  
<outputMsgFiltering info="no" trace="yes" warning="yes"/> <testWindowState height="496"  
splitterClosed="No" splitterXPos="400" width="907"/> </mobileSettings>"
```

### Zie ook

[document.setMobileSettings\(\)](#)

## document.getPlayerVersion()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`document.getPlayerVersion()`

### Parameters

Geen.

### Geretourneerde waarde

Een tekenreeks die de versie van Flash Player bevat die is opgegeven met `document.setPlayerVersion()`. Als er geen waarde is ingesteld, wordt de waarde geretourneerd die is opgegeven in het dialoogvenster Publicatie-instellingen.

### Beschrijving

Methode; hiermee wordt een tekenreeks geretourneerd die de doelversie van de player voor het opgegeven document bevat. Zie [document.setPlayerVersion\(\)](#) voor een lijst met waarden die met deze methode kunnen worden geretourneerd.

Gebruik [document.asVersion](#) om de doelversie van ActionScript in het opgegeven bestand te bepalen.

### Voorbeeld

In het volgende voorbeeld worden opgegeven doelversies van de player voor het huidige document bepaald en worden die waarden opgehaald:

```
fl.getDocumentDOM().setPlayerVersion("6");  
var version = fl.getDocumentDOM().getPlayerVersion();  
fl.trace(version) // displays "6"  
fl.getDocumentDOM().setPlayerVersion("FlashPlayer10");  
var version = fl.getDocumentDOM().getPlayerVersion();  
fl.trace(version) // displays "FlashPlayer10"
```

#### Zie ook

[document.setPlayerVersion\(\)](#)

## document.getSelectionRect()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

`document.getSelectionRect()`

#### Parameters

Geen.

#### Geretourneerde waarde

Het kader van de huidige selectie of 0 als er niets is geselecteerd. Zie [document.addNewRectangle\(\)](#) voor informatie over de notatie van de geretourneerde waarde.

#### Beschrijving

Methode; hiermee wordt het kader van de huidige selectie opgehaald. Als een selectie niet rechthoekig is, wordt de kleinste rechthoek geretourneerd die om de volledige selectie past. De rechthoek is gebaseerd op de documentruimte of, in de bewerkingsmodus, op het registratiepunt (ook *oorsprong* of *nulpunt* genoemd) van het symbool dat wordt bewerkt.

#### Voorbeeld

In het volgende voorbeeld wordt het kader voor de huidige selectie opgehaald en worden de eigenschappen daarvan weergegeven:

```
var newRect = fl.getDocumentDOM().getSelectionRect();  
var outputStr = "left: " + newRect.left + " top: " + newRect.top + " right: " + newRect.right  
+ " bottom: " + newRect.bottom;  
alert(outputStr);
```

#### Zie ook

[document.selection](#), [document.setSelectionRect\(\)](#)



## document.getTextString()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getTextString([startIndex [, endIndex]])
```

### Parameters

**startIndex** Een geheel getal dat de index is van het eerste teken dat moet worden opgehaald. Deze parameter is optioneel.

**endIndex** Een geheel getal dat de index is van het laatste teken dat moet worden opgehaald. Deze parameter is optioneel.

### Geretourneerde waarde

Een tekenreeks die de geselecteerde tekst bevat.

### Beschrijving

Methode; hiermee wordt de geselecteerde tekst opgehaald. Als de optionele parameters niet worden doorgegeven, wordt de huidige tekstselectie gebruikt. Als er geen tekst is geopend om te bewerken, wordt de hele tekst geretourneerd. Als alleen *startIndex* wordt doorgegeven, wordt de tekenreeks geretourneerd van die index tot het einde van het veld. Als *startIndex* en *endIndex* worden doorgegeven, wordt de tekenreeks geretourneerd van *startIndex* tot aan, maar niet inclusief, *endIndex*.

Als er meerdere tekstvelden zijn geselecteerd, worden alle tekenreeksen samengevoegd tot een tekenreeks die wordt geretourneerd.

### Voorbeeld

In het volgende voorbeeld wordt de tekenreeks in de geselecteerde tekstvelden opgehaald:

```
fl.getDocumentDOM().getTextString();
```

In het volgende voorbeeld wordt de tekenreeks op tekenindex 5 in de geselecteerde tekstvelden opgehaald:

```
fl.getDocumentDOM().getTextString(5);
```

In het volgende voorbeeld wordt de tekenreeks opgehaald van tekenindex 2 tot aan, maar niet inclusief, tekenindex 10:

```
fl.getDocumentDOM().getTextString(2, 10);
```

### Zie ook

[document.setTextString\(\)](#)

## document.getTimeline()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getTimeline()
```

### Parameters

Geen.

### Geretourneerde waarde

Het huidige object Timeline.

### Beschrijving

Methode; hiermee wordt het huidige [Object Timeline](#) in het document opgehaald. De huidige tijdlijn kan de huidige scène, het huidige symbool dat wordt bewerkt of het huidige scherm zijn.

### Voorbeeld

In het volgende voorbeeld wordt het object Timeline opgehaald en wordt het aantal frames in de langste laag geretourneerd:

```
var longestLayer = fl.getDocumentDOM().getTimeline().frameCount;  
fl.trace("The longest layer has" + longestLayer + "frames");
```

In het volgende voorbeeld wordt de modus Op plaats bewerken geactiveerd voor het geselecteerde symbool in het werkgebied en wordt een frame ingevoegd op de tijdlijn van het symbool.

```
fl.getDocumentDOM().enterEditMode("inPlace");  
fl.getDocumentDOM().getTimeline().insertFrames();
```

In het volgende voorbeeld wordt object Timeline opgehaald en de naam van dat object weergegeven:

```
var timeline = fl.getDocumentDOM().getTimeline();  
alert(timeline.name);
```

### Zie ook

[document.currentTimeline](#), [document.timelines](#), [symbolItem.timeline](#)

## document.getTransformationPoint()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.getTransformationPoint()
```

### Parameters

Geen.

### Geretourneerde waarde

Een punt (bijvoorbeeld {x:10, y:20}, waarbij x en y drijvende-kommawaarden zijn) dat de positie van het transformatiepunt (ook *oorsprong* of *nulpunt* genoemd) aangeeft in het coördinatensysteem van het geselecteerde element.

### Beschrijving

Methode; hiermee wordt de locatie opgehaald van het transformatiepunt van de huidige selectie. U kunt het transformatiepunt gebruiken voor bewerkingen zoals roteren en scheeftrekken.

**Opmerking:** *Transformatiepunten zijn relatief ten opzichte van verschillende locaties, afhankelijk van het type item dat is geselecteerd. Zie `document.setTransformationPoint()` voor meer informatie.*

### Voorbeeld

In het volgende voorbeeld wordt het transformatiepunt voor de huidige selectie opgehaald. De eigenschap `transPoint.x` bevat de *x*-coördinaat van het transformatiepunt. De eigenschap `transPoint.y` bevat de *y*-coördinaat van het transformatiepunt.

```
var transPoint = fl.getDocumentDOM().getTransformationPoint();
```

### Zie ook

[document.setTransformationPoint\(\)](#), [element.getTransformationPoint\(\)](#)

## document.group()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.group()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie geconverteerd naar een groep.

### Voorbeeld

In het volgende voorbeeld worden de objecten in de huidige selectie geconverteerd naar een groep:

```
fl.getDocumentDOM().group();
```

### Zie ook

[document.unGroup\(\)](#)

## document.height

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.height`

### Beschrijving

Eigenschap; een geheel getal dat de hoogte van het document (werkgebied) in pixels aangeeft.

### Voorbeeld

In het volgende voorbeeld wordt de hoogte van het werkgebied ingesteld op 400 pixels:

```
fl.getDocumentDOM().height = 400;
```

### Zie ook

[document.width](#)

## document.id

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`document.id`

### Beschrijving

Alleen-lezen eigenschap; een uniek geheel getal (automatisch toegewezen) dat een document tijdens een Flash-sessie identificeert. Gebruik deze eigenschap samen met `fl.findDocumentDOM()` om een bepaald document op te geven voor een actie.

### Voorbeeld

In het volgende voorbeeld wordt de document-id voor het huidige document weergegeven:

```
fl.trace("Current doc's internal ID is: " + fl.getDocumentDOM().id);
```

### Zie ook

[fl.findDocumentDOM\(\)](#)

## document.importFile()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.importFile(fileURI [, importToLibrary])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad van het te importeren bestand bevat.

**importToLibrary** Een Booleaanse waarde die aangeeft of het bestand alleen moet worden geïmporteerd in de bibliotheek van het document (`true`) of dat ook een kopie in het werkgebied moet worden geplaatst (`false`). De standaardwaarde is `false`.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een bestand naar een document geïmporteerd. Deze methode voert dezelfde bewerking uit als de menuopdracht Naar bibliotheek importeren of Naar werkgebied importeren. Als u een publicatieprofiel wilt importeren, gebruikt u `document.importPublishProfile()`.

#### Voorbeeld

In het volgende voorbeeld kan de gebruiker een bestand zoeken dat in het werkgebied moet worden geïmporteerd:

```
var dom = fl.getDocumentDOM();  
var URI = fl.browseForFileURL("select", "Import File");  
dom.importFile(URI);
```

#### Zie ook

`document.importSWF()`, `fl.browseForFileURL()`

## document.importPublishProfile()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.importPublishProfile( fileURI )
```

#### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad bevat van het XML-bestand waarin het te importeren profiel is gedefinieerd.

#### Geretourneerde waarde

Een geheel getal dat de index is van het geïmporteerde profiel in de lijst met profielen. Retourneert -1 als het profiel niet kan worden geïmporteerd.

#### Beschrijving

Methode; hiermee wordt een profiel uit een bestand geïmporteerd.

#### Voorbeeld

In het volgende voorbeeld wordt het profiel geïmporteerd dat in het bestand Profile.xml aanwezig is en wordt de bijbehorende index in de lijst met profielen weergegeven:

```
alert(fl.getDocumentDOM().importPublishProfile('file:///C:/Documents and  
Settings/janeUser/Desktop/profile.xml'));
```

## document.importPublishProfileString()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.importPublishProfileString(xmlString)
```

### Parameters

**xmlString** Een tekenreeks die de XML-gegevens bevat die als het huidige profiel moeten worden geïmporteerd.

### Geretourneerde waarde

De Booleaanse waarde `true` als de tekenreeks is geïmporteerd, anders `false`.

### Beschrijving

Methode hiermee wordt een XML-tekenreeks die een publicatieprofiel bevat geïmporteerd en ingesteld als het huidige profiel. Als u een XML-tekenreeks wilt genereren om te importeren, gebruikt u [document.exportPublishProfileString\(\)](#) voordat u deze methode gebruikt.

### Voorbeeld

In het volgende voorbeeld wordt het standaardprofiel geëxporteerd als een XML-tekenreeks. De standaard JavaScript-opdracht `replace` wordt gebruikt om de XML-tekenreeks te wijzigen. De tekenreeks wordt vervolgens geïmporteerd en de standaardinstelling voor ActionScript 3-uitvoer wordt ingesteld op ActionScript 1.

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString('Default');
fl.trace(profileXML);
var newProfileXML = profileXML.replace("<ActionScriptVersion>3</ActionScriptVersion>",
"<ActionScriptVersion>1</ActionScriptVersion>");
fl.getDocumentDOM().importPublishProfileString(newProfileXML);
```

## document.importSWF()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.importSWF(fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het SWF-bestand bevat dat moet worden geïmporteerd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een SWF-bestand in een document geïmporteerd. Deze methode voert dezelfde bewerking uit als het opgeven van een SWF-bestand met de menuopdracht Importeren. In Flash 8 en hoger kunt u ook `document.importFile()` gebruiken om een SWF-bestand (en andere typen bestanden) te importeren.

### Voorbeeld

In het volgende voorbeeld wordt het bestand "mySwf.swf" geïmporteerd uit de configuratiemap van Flash:

```
fl.getDocumentDOM().importSWF(fl.configURI+"mySwf.swf");
```

### Zie ook

[document.importFile\(\)](#)

## document.intersect()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.intersect()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee wordt een tekenobject gemaakt dat een doorsnede is van alle geselecteerde tekenobjecten. Deze methode retourneert `false` als er geen tekenobjecten zijn geselecteerd of als een of meer geselecteerde items geen tekenobjecten zijn.

### Voorbeeld

In het volgende voorbeeld wordt een tekenobject gemaakt dat een doorsnede is van alle geselecteerde tekenobjecten.

```
fl.getDocumentDOM().intersect();
```

### Zie ook

[document.crop\(\)](#), [document.deleteEnvelope\(\)](#), [document.punch\(\)](#), [document.union\(\)](#),  
[shape.isDrawingObject](#)

## document.library

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.library`

### Beschrijving

Alleen-lezen eigenschap; het [Object library](#) voor een document.

### Voorbeeld

In het volgende voorbeeld wordt de bibliotheek voor het actieve document opgehaald:

```
var myCurrentLib = fl.getDocumentDOM().library;
```

Ervan uitgaande dat het actieve document niet `fl.documents[1]` is, wordt in het volgende voorbeeld de bibliotheek opgehaald voor een bibliotheek die niet de focus heeft, of voor een bibliotheek die u met Bestand > Openen hebt geopend als externe bibliotheek:

```
var externalLib = fl.documents[1].library;
```

## document.libraryPath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`document.libraryPath`

### Beschrijving

Eigenschap; een tekenreeks die een lijst bevat met items in het ActionScript 3.0-bibliotheekpad van het document, dat de locatie aangeeft van SWC-bestanden of mappen die SWC-bestanden bevatten. Items in de tekenreeks worden gescheiden door puntkomma's. In het ontwerp gereedschap worden de items opgegeven door Bestand > Publicatie-instellingen te kiezen en vervolgens ActionScript 3.0-instellingen te kiezen op het tabblad Flash.

### Voorbeeld

In het volgende voorbeeld wordt de map `../Files` toegevoegd aan het bibliotheekpad van het document en wordt vervolgens het bibliotheekpad weergegeven in het deelvenster Uitvoer:

```
var myDoc = fl.getDocumentDOM();
fl.trace(myDoc.libraryPath);
myDoc.libraryPath = "../Files;" + myDoc.libraryPath;
fl.trace(myDoc.libraryPath);
```

### Zie ook

[document.externalLibraryPath](#), [document.sourcePath](#), [fl.libraryPath](#)

## document.livePreview

### Beschikbaarheid

Flash MX 2004.



### Gebruik

```
document.livePreview
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of Live voorvertoning is ingeschakeld. Als deze eigenschap op `true` is ingesteld, worden componenten in het werkgebied weergegeven zoals ze worden weergegeven in de gepubliceerde Flash-inhoud en wordt de grootte bij benadering weergegeven. Als deze eigenschap op `false` is ingesteld, worden componenten alleen als omtrekken weergegeven. De standaardwaarde is `true`.

### Voorbeeld

In het volgende voorbeeld wordt Live voorvertoning ingesteld op `false`:

```
fl.getDocumentDOM().livePreview = false;
```

## document.loadCuepointXML()

### Beschikbaarheid

Flash Professional CS5

### Gebruik

```
document.loadCuepointXML(String URI)
```

### Parameters

**URI** Tekenreeks, het absolute pad naar het XML-actiepuntenbestand.

### Beschrijving

Methode, hiermee wordt een XML-actiepuntbestand geladen. De indeling en DTD van het XML-bestand komen overeen met de indeling en DTD die door de Eigenschapcontrole voor actiepunten worden geïmporteerd en geëxporteerd. De retourwaarde komt overeen met de tekenreeks die is geserialiseerd in de eigenschap Cue Point van het object met de instantie van een FLVPlayback-component.

### Voorbeeld

In het volgende voorbeeld bevindt het XML-actiepuntenbestand zich op `C:\\testCuePoints.xml`:

```
var cuePoints = fl.getDocumentDOM().LoadCuepointXML("c:\\testCuePoints.xml");
```

## document.match()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.match(bWidth, bHeight [, bUseDocumentBounds])
```

### Parameters

**width** Een Booleaanse waarde; als de waarde is ingesteld op `true`, worden alle geselecteerde items even breed gemaakt.

**height** Een Booleaanse waarde; als de waarde is ingesteld op `true`, worden alle geselecteerde items even hoog gemaakt.

**useDocumentBounds** Een Booleaanse waarde; als de waarde is ingesteld op `true`, wordt de grootte van de objecten aangepast aan de grenzen van het document. Anders worden de grenzen van het grootste object gebruikt. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle geselecteerde objecten even groot gemaakt.

### Voorbeeld

In het volgende voorbeeld wordt alleen de breedte van de geselecteerde objecten gelijk gemaakt:

```
fl.getDocumentDOM().match(true, false);
```

In het volgende voorbeeld wordt alleen de hoogte gelijk gemaakt:

```
fl.getDocumentDOM().match(false, true);
```

In het volgende voorbeeld wordt alleen de breedte gelijk gemaakt aan de grenzen van het document:

```
fl.getDocumentDOM().match(true, false, true);
```

### Zie ook

[document.getAlignToDocument\(\)](#), [document.setAlignToDocument\(\)](#)

## document.mouseClick()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.mouseClick(position, bToggleSel, bShiftSel)
```

### Parameters

**position** Een paar drijvende-kommawaarden waarmee de *x*- en *y*-coördinaat van de klik in pixels worden opgegeven.

**bToggleSel** Een Booleaanse waarde die de status van de Shift-toets aangeeft: `true` als de toets is ingedrukt; `false` als de toets niet is ingedrukt.

**bShiftSel** Een Booleaanse waarde die de status van de voorkeur van de toepassing voor Shift-selectie aangeeft: `true` als dit is ingeschakeld; `false` als dit is uitgeschakeld.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een muisklik uitgevoerd met het gereedschap Selectie.

#### Voorbeeld

In het volgende voorbeeld wordt een muisklik uitgevoerd op de opgegeven locatie:

```
fl.getDocumentDOM().mouseClick({x:300, y:200}, false, false);
```

#### Zie ook

[document.mouseDbClick\(\)](#)

## document.mouseDbClick()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.mouseDbClick(position, bAltDown, bShiftDown, bShiftSelect)
```

#### Parameters

**position** Een paar drijvende-kommawaarden waarmee de *x*- en *y*-coördinaat van de klik in pixels worden opgegeven.

**bAltDown** Een Booleaanse waarde die vastlegt of de Alt-toets was ingedrukt op het moment van de gebeurtenis: *true* voor ingedrukt; *false* voor niet ingedrukt.

**bShiftDown** Een Booleaanse waarde die vastlegt of de Shift-toets was ingedrukt op het moment van de gebeurtenis: *true* voor ingedrukt; *false* voor niet ingedrukt.

**bShiftSelect** Een Booleaanse waarde die de status van de voorkeur van de toepassing voor Shift-selectie aangeeft: *true* voor ingeschakeld; *false* voor uitgeschakeld.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een dubbele muisklik uitgevoerd met het gereedschap Selectie.

#### Voorbeeld

In het volgende voorbeeld wordt een dubbele muisklik uitgevoerd op de opgegeven locatie:

```
fl.getDocumentDOM().mouseDbClick({x:392.9, y:73}, false, false, true);
```

#### Zie ook

[document.mouseClick\(\)](#)

## document.moveSelectedBezierPointsBy()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.moveSelectedBezierPointsBy(delta)
```

### Parameters

**delta** Een paar drijvende-kommawaarden waarmee de  $x$ - en  $y$ -coördinaat in pixels worden opgegeven waarmee de geselecteerde Bézier-punten worden verplaatst. Als u bijvoorbeeld  $\{x:1, y:2\}$  opgeeft, geeft u een locatie aan die zich één pixel rechts van en twee pixels onder de huidige locatie bevindt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; als de selectie ten minste één pad bevat waarop ten minste één Bézier-punt is geselecteerd, worden alle geselecteerde Bézier-punten op alle geselecteerde paden met de opgegeven hoeveelheid verplaatst.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde Bézier-punten 10 pixels naar rechts en 5 pixels naar beneden verplaatst:

```
fl.getDocumentDOM().moveSelectedBezierPointsBy({x:10, y:5});
```

## document.moveSelectionBy()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.moveSelectionBy(distanceToMove)
```

### Parameters

**distanceToMove** Een paar drijvende-kommawaarden waarmee de  $x$ - en  $y$ -coördinaat worden opgegeven waarmee de selectie wordt verplaatst. Als u bijvoorbeeld  $\{x:1, y:2\}$  opgeeft, geeft u een locatie aan die zich één pixel rechts van en twee pixels onder de huidige locatie bevindt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden geselecteerde objecten over een bepaalde afstand verplaatst.

**Opmerking:** Wanneer de gebruiker het item verplaatst met de pijltoetsen, worden alle toetsindrukken in het deelvenster Historie gecombineerd tot één verplaatsingsstap. Wanneer de gebruiker de pijltoetsen herhaaldelijk indrukt in plaats van meerdere stappen uit te voeren in het deelvenster Historie, wordt één stap uitgevoerd en worden de argumenten bijgewerkt met de herhaalde pijltoetsen.

Zie `document.setSelectionRect()`, `document.mouseClick()`, `document.mouseDb1C1k()` en het [Object Element](#) voor informatie over het maken van een selectie.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde item 62 pixels naar rechts en 84 pixels naar beneden verplaatst:

```
fl.getDocumentDOM().moveSelectionBy({x:62, y:84});
```

## document.name

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.name
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks met de naam van een document (FLA-bestand).

### Voorbeeld

In het volgende voorbeeld wordt de variabele `fileName` ingesteld op de bestandsnaam van het eerste document in de documentenarray:

```
var fileName = flash.documents[0].name;
```

In het volgende voorbeeld worden de namen van alle geopende documenten weergegeven in het deelvenster Uitvoer:

```
var openDocs = fl.documents;  
for(var i=0;i < openDocs.length; i++){  
    fl.trace(i + " " + openDocs[i].name + "\n");  
}
```

## document.optimizeCurves()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.optimizeCurves(smoothing, bUseMultiplePasses)
```

### Parameters

**smoothing** Een geheel getal in het bereik van 0 tot en met 100, waarbij 0 betekent niet vloeiend maken en 100 betekent maximaal vloeiend maken.

**useMultiplePasses** Een Booleaanse waarde; als de waarde is ingesteld op `true`, moet de methode meerdere stappen gebruiken. Dit is langzamer maar geeft een beter resultaat. Deze parameter heeft hetzelfde effect als klikken op de knop Meerdere stappen uitvoeren in het dialoogvenster Curven optimaliseren.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de huidige selectie optimaal vloeiend gemaakt; hiervoor mogen meerdere stappen worden uitgevoerd als dat is aangegeven. Deze methode is hetzelfde als het selecteren van Wijzigen > Vorm > Optimaliseren.

#### Voorbeeld

In het volgende voorbeeld wordt de curve van de huidige selectie 50° vloeiend gemaakt in meerdere stappen:

```
fl.getDocumentDOM().optimizeCurves(50, true);
```

## document.path

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.path
```

#### Beschrijving

Alleen-lezen eigenschap; een tekenreeks met het pad van het document in een platformspecifieke notatie. Als het document nog nooit is opgeslagen, is de waarde van deze eigenschap `undefined`.

#### Voorbeeld

In het volgende voorbeeld wordt het pad van het eerste document in de documentenarray weergegeven in het deelvenster Uitvoer. U moet het document opslaan voordat u dit script kunt uitvoeren. In het voorbeeld is de naam van het bestand Test.fla en wordt dit bestand opgeslagen in de map Mijn documenten op een Windows-computer.

```
var filePath = flash.documents[0].path;  
fl.trace(filePath);  
// displays C:\Documents and Settings\\My Documents\test.fla
```

#### Zie ook

[document.pathURI](#)

## document.pathURI

#### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`document.pathURI`

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks, uitgedrukt als bestand:/// URI, die het pad van het document vertegenwoordigt. Als het document nog nooit is opgeslagen, is de waarde van deze eigenschap `undefined`.

### Voorbeeld

In het volgende voorbeeld wordt het pad van het eerste document in de documentenarray als een tekenreeks bestand:/// URI weergegeven in het deelvenster Uitvoer. U moet het document opslaan voordat u dit script kunt uitvoeren. In het voorbeeld is de naam van het bestand `Test.fla` en wordt dit bestand opgeslagen in de map Mijn documenten op een Windows-computer.

```
var filePathURI = flash.documents[0].pathURI;
fl.trace(filePathURI);
// displays file:///C:/Documents%20and%20Settings/<userName>/My%20Documents/test.fla
```

### Zie ook

[document.path](#)

## document.publish()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.publish()`

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het document gepubliceerd met de actieve publicatie-instellingen (Bestand > Publicatie-instellingen). Deze methode is gelijk aan het selecteren van Bestand > Publiceren.

### Voorbeeld

In het volgende voorbeeld wordt het huidige document gepubliceerd:

```
fl.getDocumentDOM().publish();
```

## document.publishProfiles

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.publishProfiles
```

### Beschrijving

Alleen-lezen eigenschap; een array met de namen van de publicatieprofielen voor het document.

### Voorbeeld

In het volgende voorbeeld worden de namen van de publicatieprofielen voor het document weergegeven:

```
var myPubProfiles = fl.getDocumentDOM().publishProfiles;
for (var i=0; i < myPubProfiles.length; i++){
    fl.trace(myPubProfiles[i]);
}
```

## document.punch()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.punch()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee wordt het bovenste geselecteerde tekenobject geperforeerd door alle onderliggende geselecteerde tekenobjecten. Deze methode retourneert `false` als er geen tekenobjecten zijn geselecteerd of als een of meer geselecteerde items geen tekenobjecten zijn.

### Voorbeeld

In het volgende voorbeeld wordt geperforeerd door tekenobjecten onder het geselecteerde tekenobject:

```
fl.getDocumentDOM().punch();
```

### Zie ook

[document.crop\(\)](#), [document.deleteEnvelope\(\)](#), [document.intersect\(\)](#), [document.union\(\)](#),  
[shape.isDrawingObject](#)



## document.removeAllFilters()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.removeAllFilters()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle filters van een of meer geselecteerde objecten verwijderd.

### Voorbeeld

In het volgende voorbeeld worden alle filters verwijderd van een of meer geselecteerde objecten.

```
fl.getDocumentDOM().removeAllFilters();
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableAllFilters\(\)](#), [document.getFilters\(\)](#), [document.removeFilter\(\)](#), [Object Filter](#)

## document.removeDataFromDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.removeDataFromDocument(name)
```

### Parameters

**name** Een tekenreeks waarmee de naam wordt opgegeven van de gegevens die moeten worden verwijderd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden permanente gegevens met de opgegeven naam die aan het document zijn gekoppeld, verwijderd.

### Voorbeeld

In het volgende voorbeeld worden de permanente gegevens met de naam "myData" uit het document verwijderd:

```
fl.getDocumentDOM().removeDataFromDocument("myData");
```

#### Zie ook

[document.addDataToDocument\(\)](#), [document.documentHasData\(\)](#), [document.getDataFromDocument\(\)](#)

## document.removeDataFromSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.removeDataFromSelection(name)
```

### Parameters

**name** Een tekenreeks met de naam van de permanente gegevens die moeten worden verwijderd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden permanente gegevens met de opgegeven naam verwijderd die aan de selectie zijn gekoppeld.

### Voorbeeld

In het volgende voorbeeld worden de permanent gegevens met de naam "myData" uit de selectie verwijderd:

```
fl.getDocumentDOM().removeDataFromSelection("myData");
```

#### Zie ook

[document.addDataToSelection\(\)](#)

## document.removeFilter()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.removeFilter(filterIndex)
```

### Parameters

**filterIndex** Een geheel getal dat de op nul gebaseerde index aangeeft van het filter dat uit een of meer geselecteerde objecten moet worden verwijderd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het opgegeven filter verwijderd uit de lijst met filters van een of meer geselecteerde objecten.

### Voorbeeld

In het volgende voorbeeld wordt het eerste filter (index 0) verwijderd uit de lijst met filters van een of meer geselecteerde objecten:

```
fl.getDocumentDOM().removeFilter(0);
```

### Zie ook

[document.addFilter\(\)](#), [document.changeFilterOrder\(\)](#), [document.disableFilter\(\)](#),  
[document.getFilters\(\)](#), [document.removeAllFilters\(\)](#), [Object Filter](#)

## document.renamePublishProfile()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.renamePublishProfile([profileNewName])
```

### Parameters

**profileNewName** Een optionele parameter met de nieuwe naam voor het profiel. De nieuwe naam moet uniek zijn. Als u geen naam opgeeft, wordt een standaardnaam gebruikt.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de naam is gewijzigd, anders `false`.

### Beschrijving

Methode; hiermee wordt de naam van het huidige profiel gewijzigd.

### Voorbeeld

In het volgende voorbeeld wordt de naam van het huidige profiel gewijzigd in een standaardnaam en weergegeven:

```
alert(fl.getDocumentDOM().renamePublishProfile());
```

## document.renameScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.renameScene(name)
```

### Parameters

**name** Een tekenreeks die de nieuwe naam van de scène bevat.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de naam is gewijzigd, anders `false`. Als de nieuwe naam bijvoorbeeld niet uniek is, wordt `false` geretourneerd.

### Beschrijving

Methode; hiermee wordt de naam gewijzigd van de geselecteerde scène in het deelvenster Scènes. De nieuwe naam voor de geselecteerde scène moet uniek zijn.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de huidige scène gewijzigd in "new name":

```
var success = fl.getDocumentDOM().renameScene("new name");
```

## document.reorderScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.reorderScene(sceneToMove, sceneToPutItBefore)
```

### Parameters

**sceneToMove** Een geheel getal dat aangeeft welke scène moet worden verplaatst, waarbij 0 (nul) de eerste scène is.

**sceneToPutItBefore** Een geheel getal dat aangeeft vóór welke scène de scène die met *sceneToMove* wordt opgegeven, moet worden geplaatst. Geef 0 (nul) op voor de eerste scène. Als u bijvoorbeeld 1 opgeeft voor *sceneToMove* en 0 voor *sceneToPutItBefore*, wordt de tweede scène vóór de eerste scène geplaatst. Geef -1 op om de scène naar het einde te verplaatsen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de opgegeven scène vóór een andere opgegeven scène geplaatst.

### Voorbeeld

In het volgende voorbeeld wordt de tweede scène vóór de eerste scène geplaatst:

```
fl.getDocumentDOM().reorderScene(1, 0);
```

## document.resetOvalObject()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.resetOvalObject()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle waarden in Eigenschapcontrole ingesteld op standaardinstellingen voor ovaal object. Als er objecten van het type Oval zijn geselecteerd, worden de eigenschappen van die objecten ook teruggezet op de standaardwaarden.

### Voorbeeld

In het volgende voorbeeld worden de eigenschappen voor objecten van het type Oval in het huidige document teruggezet op standaardwaarden:

```
fl.getDocumentDOM().resetOvalObject();
```

### Zie ook

[document.resetRectangleObject\(\)](#)

## document.resetRectangleObject()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.resetRectangleObject()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle waarden in Eigenschapcontrole ingesteld op de standaardinstellingen van het object Rectangle. Als er objecten van het type Rectangle zijn geselecteerd, worden de eigenschappen van die objecten ook teruggezet op de standaardwaarden.

### Voorbeeld

In het volgende voorbeeld worden de eigenschappen voor objecten van het type Rectangle in het huidige document teruggezet op standaardwaarden:

```
fl.getDocumentDOM().resetRectangleObject();
```

### Zie ook

[document.resetOvalObject\(\)](#)

## document.resetTransformation()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.resetTransformation();
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de transformatiematrix opnieuw ingesteld. Deze methode is hetzelfde als de opdracht Wijzigen > Transformeren > Transformatie verwijderen.

### Voorbeeld

In het volgende voorbeeld wordt de transformatiematrix voor de huidige selectie opnieuw ingesteld.

```
fl.getDocumentDOM().resetTransformation();
```

## document.revert()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.revert();
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de laatst opgeslagen versie van het opgegeven document hersteld. Deze methode is gelijk aan het selecteren van Bestand > Omkeren.

### Voorbeeld

In het volgende voorbeeld wordt de laatst opgeslagen versie van het huidige document hersteld:

```
fl.getDocumentDOM().revert();
```

### Zie ook

[document.canRevert\(\)](#), [fl.revertDocument\(\)](#)

## document.rotate3DSelection()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.rotate3DSelection(xyzCoordinate, bGlobalTransform)
```

### Parameters

**xyzCoordinate** Een XYZ-coördinaat die de as voor 3D-rotatie aangeeft.

**bGlobalTransform** Een Booleaanse waarde die opgeeft of de transformatiemodus globaal (`true`) of lokaal (`false`) is.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode: hiermee wordt een 3D-rotatie toegepast op de selectie. Deze methode is alleen beschikbaar voor filmclips.

### Voorbeeld

In het volgende voorbeeld wordt de selectie eerst geroteerd ten opzichte van het werkgebied (globaal) en vervolgens ten opzichte van zichzelf (lokaal).

```
var myDocument = fl.getDocumentDOM();  
myDocument.rotate3DSelection({x:52.0, y:0, z:0}, true);  
myDocument.rotate3DSelection({x:52.0, y:0, z:-55.2}, false);
```

## document.rotateSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.rotateSelection(angle [, rotationPoint])
```

### Parameters

**angle** Een drijvende-kommawaarde waarmee de hoek van de rotatie wordt opgegeven.

**rotationPoint** Een tekenreeks waarmee de zijde van het kader wordt aangegeven die moet worden geroteerd. Acceptabele waarden zijn "top right", "top left", "bottom right", "bottom left", "top center", "right center", "bottom center" en "left center". Als u geen waarde opgeeft, wordt het transformatiepunt gebruikt. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie geroteerd met een opgegeven aantal graden. Het effect is hetzelfde als het roteren van het object met het gereedschap Vrije transformatie.

### Voorbeeld

In het volgende voorbeeld wordt de selectie 45° om het transformatiepunt geroteerd:

```
fl.getDocumentDOM().rotateSelection(45);
```

In het volgende voorbeeld wordt de selectie 45° om de linkerbenenhoek geroteerd:

```
fl.getDocumentDOM().rotateSelection(45, "bottom left");
```

## document.save()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.save([bOkToSaveAs])
```

### Parameters

**bOkToSaveAs** Een optionele parameter: als de waarde `true` is of wordt weggelaten en het bestand nog nooit is opgeslagen, wordt het dialoogvenster Opslaan als geopend. Als de waarde `false` is en het bestand nog nooit is opgeslagen, wordt het bestand niet opgeslagen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het bestand is opgeslagen, anders `false`.

### Beschrijving

Methode; hiermee wordt het document op de standaardlocatie opgeslagen. Deze methode is gelijk aan het selecteren van Bestand > Opslaan.

Als u een naam voor het bestand wilt opgeven (in plaats van het op te slaan met dezelfde naam), gebruikt u `fl.saveDocument()`.

**Opmerking:** Als het bestand nieuw is en nog niet is gewijzigd of opgeslagen of als het bestand niet is gewijzigd sinds het de laatste keer is opgeslagen, heeft deze methode geen effect en wordt `false` geretourneerd. Als u een niet-opgeslagen of niet-gewijzigd bestand wilt opslaan, gebruikt u `document.saveAndCompact()` of `fl.saveDocumentAs()`.



### Voorbeeld

In het volgende voorbeeld wordt het huidige document opgeslagen op de standaardlocatie:

```
fl.getDocumentDOM().save();
```

### Zie ook

[document.saveAndCompact\(\)](#), [fl.saveAll\(\)](#), [fl.saveDocument\(\)](#), [fl.saveDocumentAs\(\)](#)

## document.saveAndCompact()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.saveAndCompact([bOkToSaveAs])
```

### Parameters

**bOkToSaveAs** Een optionele parameter: als de waarde `true` is of wordt weggelaten en het bestand nog nooit is opgeslagen, wordt het dialoogvenster Opslaan als geopend. Als de waarde `false` is en het bestand nog nooit is opgeslagen, wordt het bestand niet opgeslagen. De standaardwaarde is `true`.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het bestand is opgeslagen en gecomprimeerd, anders `false`.

### Beschrijving

Methode; hiermee wordt het bestand opgeslagen en gecomprimeerd. Deze methode is gelijk aan het selecteren van Bestand > Opslaan en comprimeren.

**Opmerking:** Als het bestand nog nooit is opgeslagen, wordt `true` geretourneerd, zelfs als de gebruiker het dialoogvenster Opslaan als annuleert. Als u wilt weten of het bestand echt is opgeslagen, gebruikt u [fl.saveDocumentAs\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt het huidige document opgeslagen en gecomprimeerd:

```
fl.getDocumentDOM().saveAndCompact();
```

### Zie ook

[document.save\(\)](#), [fl.saveDocumentAs\(\)](#), [fl.saveDocument\(\)](#), [fl.saveAll\(\)](#)

## document.scaleSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.scaleSelection(xScale, yScale [, whichCorner])
```

### Parameters

**xScale** Een drijvende komma-waarde waarmee de waarde van *x* wordt opgegeven waarmee moet worden geschaald.

**yScale** Een drijvende komma-waarde waarmee de waarde van *y* wordt opgegeven waarmee moet worden geschaald.

**whichCorner** Een tekenreeks die aangeeft ten opzichte van welke rand de transformatie plaatsvindt. Als u deze waarde weglaat, wordt ten opzichte van het transformatiepunt geschaald. Acceptabele waarden zijn: "bottom left", "bottom right", "top right", "top left", "top center", "right center", "bottom center" en "left center". Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie met een bepaalde hoeveelheid geschaald. Deze methode is hetzelfde als het schalen van het object met het gereedschap Vrije transformatie.

### Voorbeeld

In het volgende voorbeeld wordt de breedte van de huidige selectie verdubbeld en wordt de hoogte tot de helft verkleind:

```
fl.getDocumentDOM().scaleSelection(2.0, 0.5);
```

In het volgende voorbeeld wordt de selectie verticaal gespiegeld:

```
fl.getDocumentDOM().scaleSelection(1, -1);
```

In het volgende voorbeeld wordt de selectie horizontaal gespiegeld:

```
fl.getDocumentDOM().scaleSelection(-1, 1);
```

In het volgende voorbeeld wordt de selectie verticaal geschaald met de factor 1,9 ten opzichte van middenboven:

```
fl.getDocumentDOM().scaleSelection(1, 1.90, 'top center');
```

## document.screenOutline

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.screenOutline
```

### Beschrijving

Alleen-lezen eigenschap; het huidige object ScreenOutline voor het document. Voordat u het object de eerste keer opent, moet u met `document.allowScreens()` bepalen of de eigenschap bestaat.

### Voorbeeld

In het volgende voorbeeld wordt de array met waarden van de eigenschap `screenOutline` weergegeven:

```
var myArray = new Array();
for(var i in fl.getDocumentDOM().screenOutline) {
    myArray.push(" "+i+" : "+fl.getDocumentDOM().screenOutline[i]) ;
}
fl.trace("Here is the property dump for screenOutline: "+myArray);
```

#### Zie ook

[document.allowScreens\(\)](#)

## document.selectAll()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

`document.selectAll()`

#### Parameters

Geen.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee worden alle items in het werkgebied geselecteerd. Deze methode is gelijk aan het drukken op Ctrl+A (Windows) of Command+A (Macintosh) of het selecteren van Bewerken > Alles selecteren.

#### Voorbeeld

In het volgende voorbeeld wordt alles geselecteerd wat momenteel zichtbaar is voor de gebruiker:

```
fl.getDocumentDOM().selectAll();
```

#### Zie ook

[document.selection](#), [document.selectNone\(\)](#)

## document.selection

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

`document.selection`

### Beschrijving

Eigenschap; een array met de geselecteerde objecten in het document. Als er niet is geselecteerd, wordt een array met lengte nul geretourneerd. Als er geen document is geopend, wordt `null` geretourneerd.

Als u objecten aan de array wilt toevoegen, moet u deze eerst op een van de volgende manieren selecteren:

- Handmatig een of meer objecten selecteren in het werkgebied.
- Gebruik een selectiemethode, zoals `document.setSelectionRect()`, `document.setSelectionBounds()`, `document.mouseClick()`, `document.mouseDbClk()` of `document.selectAll()`.
- Selecteer een of meer frames handmatig.
- Gebruik een methode van het [Object Timeline](#) om een of meer frames te selecteren, zoals `timeline.getSelectedFrames()`, `timeline.setSelectedFrames()` of `timeline.selectAllFrames()`.
- Alle elementen in een bepaald frame opgeven (zie [Object Element](#)). Zie het eerste voorbeeld hieronder.
- Maak een array met een of meer elementen en wijs die array toe aan de array `document.selection`. Zie het derde voorbeeld hieronder.

### Voorbeeld

In het volgende voorbeeld worden alle elementen in Frame 11 toegewezen aan de huidige selectie (denk eraan dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().currentFrame = 10;
fl.getDocumentDOM().selection =
fl.getDocumentDOM().getTimeline().layers[0].frames[10].elements;
```

In het volgende voorbeeld wordt in de linkerbovenhoek van het werkgebied een rechthoek gemaakt en wordt onder de rechthoek een tekenreeks gemaakt. Vervolgens worden beide objecten met `document.setSelectionRect()` geselecteerd en aan de array `document.selection` toegevoegd. Als laatste wordt de inhoud van `document.selection` weergegeven in het deelvenster Uitvoer.

```
fl.getDocumentDOM().addNewRectangle({left:0, top:0, right:99, bottom:99}, 0);
fl.getDocumentDOM().addNewText({left:-1, top:117.3, right:9.2, bottom:134.6});
fl.getDocumentDOM().setTextString('Hello World');
fl.getDocumentDOM().setSelectionRect({left:-28, top:-22, right:156.0, bottom:163});
```

```
var theSelectionArray = fl.getDocumentDOM().selection;
```

```
for(var i=0;i<theSelectionArray.length;i++){
fl.trace("fl.getDocumentDOM().selection["+i+"] = " + theSelectionArray[i]);
}
```

Het volgende voorbeeld is een geavanceerd voorbeeld. Hierin wordt weergegeven hoe de array met lagen en de array met elementen worden doorlopen om instanties van een bepaald symbool te bepalen en deze te selecteren. U kunt dit voorbeeld uitbreiden voor meerdere frames of scènes. In dit voorbeeld worden alle instanties van de filmclip `myMovieClip` in het eerste frame toegewezen aan de huidige selectie:

```
// Assigns the layers array to the variable "theLayers".
var theLayers = fl.getDocumentDOM().getTimeline().layers;
// Creates an array to hold all the elements
// that are instances of "myMovieClip".
var myArray = new Array();
// Counter variable
var x = 0;
// Begin loop through all the layers.
for (var i = 0; i < theLayers.length; i++) {
    // Gets the array of elements in Frame 1
    // and assigns it to the array "theElems".
    var theElems = theLayers[i].frames[0].elements;
    // Begin loop through the elements on a layer.
    for (var c = 0; c < theElems.length; c++) {
        // Checks to see if the element is of type "instance".
        if (theElems[c].elementType == "instance") {
            // If the element is an instance, it checks
            // if it is an instance of "myMovieClip".
            if (theElems[c].libraryItem.name == "myMovieClip") {
                // Assigns elements that are instances of "myMovieClip" to "myArray".
                myArray[x] = theElems[c];
                // Increments counter variable.
                x++;
            }
        }
    }
}
// Now that you have assigned all the instances of "myMovieClip"
// to "myArray", you then set the document.selection array
// equal to myArray. This selects the objects on the Stage.
fl.getDocumentDOM().selection = myArray;
```

## document.selectNone()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.selectNone()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie van alle geselecteerde items opgeheven.

### Voorbeeld

In het volgende voorbeeld wordt de selectie van alle geselecteerde items opgeheven:

```
fl.getDocumentDOM().selectNone();
```

#### Zie ook

[document.selectAll\(\)](#), [document.selection](#)

## document.setAlignToDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setAlignToDocument(bToStage)
```

### Parameters

**bToStage** Een Booleaanse waarde: als de waarde op `true` is ingesteld, worden objecten op het werkgebied uitgelijnd. Als de waarde is ingesteld op `false`, gebeurt dit niet.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden de voorkeuren voor [document.align\(\)](#), [document.distribute\(\)](#), [document.match\(\)](#) en [document.space\(\)](#) voor het document ingesteld. Deze methode is gelijk aan het inschakelen van de knop Met werkgebied in het deelvenster Uitlijnen.

### Voorbeeld

In het volgende voorbeeld wordt de knop Met werkgebied in het deelvenster Uitlijnen ingeschakeld om objecten uit te lijnen met het werkgebied:

```
fl.getDocumentDOM().setAlignToDocument(true);
```

#### Zie ook

[document.getAlignToDocument\(\)](#)

## document.setBlendMode()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.setBlendMode(mode)
```

### Parameters

**mode** Een tekenreeks waarmee de gewenste overvloeimodus voor de geselecteerde objecten wordt opgegeven. Acceptabele waarden zijn "normal", "layer", "multiply", "screen", "overlay", "hardlight", "lighten", "darken", "difference", "add", "subtract", "invert", "alpha" en "erase".

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de overvloeimodus voor de geselecteerde objecten ingesteld.

### Voorbeeld

In het volgende voorbeeld wordt de overvloeimodus voor de geselecteerde objecten ingesteld op "add".

```
fl.getDocumentDOM().setBlendMode("add");
```

### Zie ook

[document.addFilter\(\)](#), [document.setFilterProperty\(\)](#), [symbolInstance.blendMode](#)

## document.setCustomFill()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setCustomFill(fill)
```

### Parameters

**fill** Een object Fill dat de vulinstellingen bevat die moeten worden gebruikt. Zie [Object Fill](#).

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden de vulinstellingen ingesteld voor het deelvenster Gereedschappen, Eigenschapcontrole en alle geselecteerde vormen. Zo kunt u met een script de vulinstellingen instellen voordat u het object tekent, in plaats van het object te tekenen, te selecteren en vervolgens de vulinstellingen te wijzigen. Ook kunt u met een script de vulinstellingen van het deelvenster Gereedschappen en Eigenschapcontrole wijzigen.

### Voorbeeld

In het volgende voorbeeld wordt de kleur van de vulkleurstaal in het deelvenster Gereedschappen en Eigenschapcontrole en van alle geselecteerde vormen gewijzigd in wit:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.color = '#FFFFFF';  
fill.style = "solid";  
fl.getDocumentDOM().setCustomFill(fill);
```

### Zie ook

[document.getCustomFill\(\)](#)

## document.setCustomStroke()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setCustomStroke(stroke)
```

### Parameters

**stroke** Een [Object Stroke](#).

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden de streekinstellingen ingesteld voor het deelvenster Gereedschappen, Eigenschapcontrole en alle geselecteerde vormen. Zo kunt u met een script de streekinstellingen instellen voordat u het object tekent, in plaats van het object te tekenen, te selecteren en vervolgens de streekinstellingen te wijzigen. Ook kunt u met een script de streekinstellingen van het deelvenster Gereedschappen en Eigenschapcontrole wijzigen.

### Voorbeeld

In het volgende voorbeeld wordt de strekdikte in het deelvenster Gereedschappen, in Eigenschapcontrole en van alle geselecteerde vormen gewijzigd:

```
var stroke = fl.getDocumentDOM().getCustomStroke();  
stroke.thickness += 2;  
fl.getDocumentDOM().setCustomStroke(stroke);
```

### Zie ook

[document.getCustomStroke\(\)](#)

## document.setElementProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setElementProperty(property, value)
```

### Parameters

**eigenschap** Een tekenreeks met de naam van de eigenschap `Element` die moet worden ingesteld. Zie de overzichtstabel met eigenschappen van het [Object Element](#) voor een lijst eigenschappen en waarden.



U kunt met deze methode geen waarden instellen voor alleen-lezen eigenschappen, zoals `element.elementType`, `element.top` of `element.left`.

**value** Een geheel getal dat de waarde aangeeft die moet worden ingesteld in de opgegeven eigenschap `Element`.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de opgegeven eigenschap `Element` ingesteld voor een of meer geselecteerde objecten in het document. Als er niets is geselecteerd, gebeurt er niets.

#### Voorbeeld

In het volgende voorbeeld wordt de breedte van alle geselecteerde objecten ingesteld op 100 en de hoogte op 50:

```
fl.getDocumentDOM().setElementProperty("width", 100);  
fl.getDocumentDOM().setElementProperty("height", 50);
```

## document.setElementTextAttr()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.setElementTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

#### Parameters

**attrName** Een tekenreeks met de naam van de eigenschap `TextAttrs` die moet worden gewijzigd.

**attrValue** De waarde waarop de eigenschap `TextAttrs` moet worden ingesteld. Zie de overzichtstabel met eigenschappen van het [Object TextAttrs](#) voor een lijst met namen van eigenschappen en verwachte waarden.

**startIndex** Een geheel getal dat de index is van het eerste teken waarop de bewerking betrekking heeft. Deze parameter is optioneel.

**endIndex** Een geheel getal dat de index is van het laatste teken waarop de bewerking betrekking heeft. Deze parameter is optioneel.

#### Geretourneerde waarde

Een Booleaanse waarde: `true` als de eigenschap van ten minste één tekstkenmerk is gewijzigd, anders `false`.

#### Beschrijving

Methode; hiermee wordt de opgegeven eigenschap `textAttrs` van de geselecteerde tekstitems ingesteld op de opgegeven waarde. Zie de overzichtstabel met eigenschappen van het [Object TextAttrs](#) voor een lijst met namen van eigenschappen en toegestane waarden. Als de optionele parameters niet worden doorgegeven, wordt de stijl ingesteld van de geselecteerde tekst of van het hele tekstveld als er geen tekst is geselecteerd. Als alleen `startIndex` wordt doorgegeven, worden de kenmerken van dat teken ingesteld. Als `startIndex` en `endIndex` worden doorgegeven, worden de kenmerken ingesteld voor de tekens van `startIndex` tot aan, maar niet inclusief, `endIndex`. Als alinea's worden opgegeven, heeft de bewerking betrekking op alle alinea's in het bereik.

### Voorbeeld

In de volgende voorbeelden worden de tekstenmerken `fillColor`, `italic` en `bold` ingesteld voor de geselecteerde tekstitems:

```
var success = fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00");  
var pass = fl.getDocumentDOM().setElementTextAttr("italic", true, 10);  
var ok = fl.getDocumentDOM().setElementTextAttr("bold", true, 5, 15);
```

## document.setFillColor()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setFillColor(color)
```

### Parameters

`color` De kleur van de vulling, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

Als de waarde is ingesteld op `null`, wordt geen vulkleur ingesteld. Dit is hetzelfde als het instellen van geen vulling voor de vulkleurstaal in de gebruikersinterface.

### Geretoureerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de vulkleur van de selectie gewijzigd in de opgegeven kleur. Zie

[document.setCustomFill\(\)](#) voor informatie over het wijzigen van de vulkleur in het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In de eerste drie instructies in het volgende voorbeeld wordt de vulkleur ingesteld met elk van de verschillende notaties voor het opgeven van kleur. Met de vierde instructie wordt de vulling ingesteld op geen vulling

```
fl.getDocumentDOM().setFillColor("#cc00cc");  
fl.getDocumentDOM().setFillColor(0xcc00cc);  
fl.getDocumentDOM().setFillColor(120000);  
fl.getDocumentDOM().setFillColor(null);
```

## document.setFilterProperty()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.setFilterProperty(property, filterIndex, value)
```

### Parameters

**eigenschap** Een tekenreeks die de eigenschap bevat die moet worden ingesteld. Acceptabele waarden zijn "blurX", "blurY", "quality", "angle", "distance", "strength", "knockout", "inner", "bevelType", "color", "shadowColor" en "highlightColor".

**filterIndex** Een geheel getal dat de op nul gebaseerde index van het filter in de lijst met filters aangeeft.

**value** Een getal of een tekenreeks die de waarde bevat die moet worden ingesteld voor de opgegeven filtereigenschap. Acceptabele waarden zijn afhankelijk van de eigenschap en het filter die worden ingesteld.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een opgegeven filtereigenschap voor de geselecteerde objecten ingesteld (ervan uitgaande dat het object het opgegeven filter ondersteunt).

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `quality` ingesteld op 2 voor het tweede filter (index 1) in de lijst met filters van de geselecteerde objecten en wordt vervolgens de eigenschap `shadowColor` van het eerste filter in de lijst met filters ingesteld voor de geselecteerde objecten:

```
fl.getDocumentDOM().setFilterProperty("quality", 1, 2);  
fl.getDocumentDOM().setFilterProperty("shadowColor", 0, "#FF00FF");
```

### Zie ook

[document.addFilter\(\)](#), [document.getFilters\(\)](#), [document.setBlendMode\(\)](#), [document.setFilters\(\)](#), [Object Filter](#)

## document.setFilters()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.setFilters(filterArray)
```

### Parameters

**filterArray** De array met filters die momenteel zijn opgegeven.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden filters toegepast op de geselecteerde objecten. Gebruik deze methode nadat `document.getFilters()` is aangeroepen en de gewenste wijzigingen in de filters zijn aangebracht.

### Voorbeeld

In het volgende voorbeeld worden de filters voor het geselecteerde object opgehaald en wordt de eigenschap `blurX` voor alle vervagingsfilters ingesteld op 50:

```
var myFilters = fl.getDocumentDOM().getFilters();
for (i=0; i < myFilters.length; i++) {
    if (myFilters[i].name == "blurFilter"){
        myFilters[i].blurX = 50;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

[document.addFilter\(\)](#), [document.getFilters\(\)](#), [document.setFilterProperty\(\)](#), [Object Filter](#)

## document.setInstanceAlpha()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setInstanceAlpha(opacity)
```

### Parameters

`opacity` Een geheel getal in het bereik van 0 (transparant) tot en met 100 (volledig verzadigd) waarmee de transparantie van de instantie wordt aangepast.

### Geretourneerde waarde

Niets.

### Beschrijving

Methoden; hiermee wordt de dekking van de instantie ingesteld.

### Voorbeeld

In het volgende voorbeeld wordt de dekking van de tint ingesteld op de waarde 50:

```
fl.getDocumentDOM().setInstanceAlpha(50);
```

## document.setInstanceBrightness()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setInstanceBrightness(brightness)
```

### Parameters

**brightness** Een geheel getal waarmee de helderheid wordt opgegeven als een waarde in het bereik van -100 (zwart) tot en met 100 (wit).

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de helderheid van de instantie ingesteld.

### Voorbeeld

In het volgende voorbeeld wordt de helderheid van de instantie ingesteld op de waarde 50:

```
fl.getDocumentDOM().setInstanceBrightness(50);
```

## document.setInstanceTint()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setInstanceTint( color, strength )
```

### Parameters

**color** De kleur van de tint, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

**strength** Een geheel getal in het bereik van 0 tot en met 100 dat de dekking van de tint bepaalt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de tint van de instantie ingesteld.

### Voorbeeld

In het volgende voorbeeld wordt de tint voor de geselecteerde instantie ingesteld op rood met een dekkingswaarde van 50:

```
fl.getDocumentDOM().setInstanceTint(0xff0000, 50);
```

## document.setMetadata()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.setMetadata(strMetadata)
```

### Parameters

**strMetadata** Een tekenreeks die de XML-metagegevens bevat die moeten worden gekoppeld aan het document. Zie de volgende beschrijving voor meer informatie.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee worden de XML-metagegevens voor het opgegeven document ingesteld, waarbij alle bestaande metagegevens worden overschreven. De XML die als *strMetadata* wordt doorgegeven, wordt gevalideerd en kan opnieuw worden geschreven voordat deze wordt opgeslagen. Als de gegevens niet als geldige XML-gegevens kunnen worden gevalideerd of als bepaalde regels worden geschonden, worden de XML-metagegevens niet ingesteld en wordt `false` geretourneerd. (Als `false` wordt geretourneerd, is het niet mogelijk meer gedetailleerde foutgegevens te verkrijgen.)

**Opmerking:** *Zelfs als `true` wordt geretourneerd, komen de XML-gegevens die worden ingesteld, mogelijk niet exact overeen met de tekenreeks die u hebt doorgegeven. Als u de exacte waarde van de ingestelde XML-gegevens wilt ophalen, gebruikt u `document.getMetadata()`.*

Voor de metagegevens wordt de RDF-indeling gebruikt die compatibel is met de XMP-specificatie. Raadpleeg de volgende bronnen voor meer informatie over RDF en XMP:

- De RDF Primer op [www.w3.org/TR/rdf-primer/](http://www.w3.org/TR/rdf-primer/)
- De RDF-specificatie op [www.w3.org/TR/1999/REC-rdf-syntax-19990222/](http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/)
- De XMP-startpagina op [www.adobe.com/products/xmp/](http://www.adobe.com/products/xmp/)

### Voorbeeld

In de volgende voorbeelden ziet u aan aantal verschillende manieren om dezelfde gegevens weer te geven. In al deze gevallen, behalve in het tweede geval, worden de gegevens niet opnieuw geschreven (regeleinden worden wel verwijderd) als de gegevens naar `Document.setMetadata()` zijn verzonden.

In het eerste voorbeeld zijn de metagegevens opgenomen in codes, waarbij verschillende schema's in afzonderlijke `rdf:Description`-codes zijn geplaatst:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/'>
<dc:title>Simple title</dc:title>
<dc:description>Simple description</dc:description>
</rdf:Description>
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
<xmp:CreateDate>2004-10-12T10:29-07:00</xmp:CreateDate>
<xmp:CreatorTool>Flash Authoring WIN 8,0,0,215</xmp:CreatorTool>
</rdf:Description>
</rdf:RDF>
```

In het tweede voorbeeld zijn de metagegevens opgenomen in codes, maar bevinden de verschillende schema's zich in één `rdf:Description`-code: Dit voorbeeld bevat ook opmerkingen, maar deze worden genegeerd en verwijderd door `Document.setMetadata()`:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!-- This is before the first rdf:Description tag -->
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/'>
<dc:title>Simple title</dc:title>
<dc:description>Simple description</dc:description>
</rdf:Description>
  <!-- This is between the two rdf:Description tags -->
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
<xmp:CreateDate>2004-10-12T10:29-07:00</xmp:CreateDate>
<xmp:CreatorTool>Flash Authoring WIN 8,0,0,215</xmp:CreatorTool>
</rdf:Description>
  <!-- This is after the second rdf:Description tag -->
</rdf:RDF>
```

In het derde voorbeeld zijn de metagegevens opgenomen in kenmerken en zijn alle verschillende schema's in één `rdf:Description`-code opgenomen:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/' dc:title='Simple title'
dc:description='Simple description' />
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'
xmp:CreateDate='2004-10-12T10:29-07:00' xmp:CreatorTool='Flash Authoring WIN 8,0,0,215' />
</rdf:RDF>
```

### Zie ook

[document.getMetadata\(\)](#)

## document.setMobileSettings()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`document.setMobileSettings(xmlString)`

### Parameters

**xmlString** Een tekenreeks waarmee de XML-instellingen in een mobiel FLA-bestand worden beschreven.

### Geretourneerde waarde

De waarde `true` als de instellingen zijn ingesteld, anders `false`.

### Beschrijving

Methode; hiermee wordt de waarde van een tekenreeks met XML-instellingen in een mobiel FLA-bestand ingesteld. (De meeste mobiele FLA-bestanden bevatten een XML-tekenreeks waarin de instellingen in het document worden beschreven.)

### Voorbeeld

In het volgende voorbeeld wordt de tekenreeks met XML-instellingen voor een mobiel FLA-bestand ingesteld. Het onderstaande voorbeeld bestaat uit één regel code.

```
fl.getDocumentDOM().setMobileSettings("<? xml version='1.0' encoding='UTF-16' standalone='no'
?> <mobileSettings> <contentType id='standalonePlayer' name='Standalone Player'/>
<testDevices> <testDevice id='1170' name='Generic Phone' selected='yes'/> </testDevices>
<outputMsgFiltering info='no' trace='yes' warning='yes'/> <testWindowState height='496'
splitterClosed='No' splitterXPos='400' width='907'/> </mobileSettings>");
```

### Zie ook

[document.getMobileSettings\(\)](#)

## document.setOvalObjectProperty()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.setOvalObjectProperty(propertyName, value)
```

### Parameters

**propertyName** Een tekenreeks die de eigenschap bevat die moet worden ingesteld. Zie de overzichtstabel met eigenschappen van het [Object Oval](#) voor acceptabele waarden.

**value** De waarde die aan de eigenschap moet worden toegewezen. Acceptabele waarden zijn afhankelijk van de eigenschap die u opgeeft in *propertyName*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een waarde opgegeven voor een opgegeven eigenschap van primitieve ovale objecten.

### Voorbeeld

Zie de afzonderlijke eigenschappen in [Object Oval](#) voor voorbeelden.

### Zie ook

[Object Oval](#), [shape.isOvalObject](#)



## document.setPlayerVersion()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.setPlayerVersion(version)
```

### Parameters

**versie** Een tekenreeks die de doelversie van Flash Player voor het opgegeven document bevat. Geldige waarden zijn "FlashLite", "FlashLite11", "FlashLite20", "FlashLite30", "1", "2", "3", "4", "5", "6", "7", "8", "9", "FlashPlayer10" en "AdobeAIR1\_1".

### Geretourneerde waarde

De waarde `true` als de versie is ingesteld, anders `false`.

### Beschrijving

Methode; hiermee wordt de doelversie van Flash Player voor het opgegeven document ingesteld. Dit is dezelfde waarde die is ingesteld in het dialoogvenster Publicatie-instellingen.

### Voorbeeld

In het volgende voorbeeld is Flash Player 6 de doelversie voor het huidige document:

```
fl.getDocumentDOM().setPlayerVersion("6");
```

### Zie ook

[document.getPlayerVersion\(\)](#)

## document.setRectangleObjectProperty()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
document.setRectangleObjectProperty(propertyName, value)
```

### Parameters

**propertyName** Een tekenreeks die de eigenschap bevat die moet worden ingesteld. Zie de overzichtstabel met eigenschappen van het [Object Rectangle](#) voor acceptabele waarden.

**value** De waarde die aan de eigenschap moet worden toegewezen. Acceptabele waarden zijn afhankelijk van de eigenschap die u opgeeft in *propertyName*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een waarde opgegeven voor een opgegeven eigenschap van primitieve rechthoekige objecten.

### Voorbeeld

Zie de afzonderlijke eigenschappen in [Object Rectangle](#) voor voorbeelden.

### Zie ook

[Object Rectangle](#), [shape.isRectangleObject](#)

## document.setSelectionBounds()

### Beschikbaarheid

Flash MX 2004; parameter *bContactSensitiveSelection* toegevoegd in Flash 8.

### Gebruik

```
document.setSelectionBounds(boundingBox [, bContactSensitiveSelection])
```

### Parameters

**boundingRectangle** Een rechthoek die de nieuwe locatie en de grootte van de selectie aangeeft. Meer informatie over het formaat van *boundingRectangle* vindt u in [document.addNewRectangle\(\)](#).

**bContactSensitiveSelection** Een Booleaanse waarde die aangeeft of de selectiemodus Contactgevoelig is ingeschakeld (*true*) of uitgeschakeld (*false*) tijdens het selecteren van objecten. De standaardwaarde is *false*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt in één enkele bewerking de selectie verplaatst en het formaat gewijzigd.

Als u een waarde voor *bContactSensitiveSelection* doorgeeft, is deze alleen geldig voor deze methode en heeft deze geen invloed op de selectiemodus Contactgevoelig voor het document (zie [fl.contactSensitiveSelection](#)).

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie verplaatst naar 10, 20 en wordt het formaat gewijzigd in 100, 200:

```
var l = 10;  
var t = 20;  
fl.getDocumentDOM().setSelectionBounds({left:l, top:t, right:(100+l), bottom:(200+t)});
```

### Zie ook

[document.selection](#), [document.setSelectionRect\(\)](#)

# document.setSelectionRect()

## Beschikbaarheid

Flash MX 2004; parameter *bContactSensitiveSelection* toegevoegd in Flash 8.

## Gebruik

```
document.setSelectionRect(rect [, bReplaceCurrentSelection [, bContactSensitiveSelection]])
```

## Parameters

**rect** Een rechthoekig object dat wordt ingesteld als geselecteerd. Zie [document.addNewRectangle\(\)](#) voor informatie over de notatie van *rect*.

**bReplaceCurrentSelection** Een Booleaanse waarde die aangeeft of de huidige selectie wordt vervangen (*true*) of wordt toegevoegd aan de huidige selectie (*false*). De standaardwaarde is *true*.

**bContactSensitiveSelection** Een Booleaanse waarde die aangeeft of de selectiemodus Contactgevoelig is ingeschakeld (*true*) of uitgeschakeld (*false*) tijdens het selecteren van objecten. De standaardwaarde is *false*.

## Geretourneerde waarde

Niets.

## Beschrijving

Methode; hiermee wordt een rechthoekig selectiekader getekend relatief ten opzichte van het werkgebied, met de opgegeven coördinaten. Dit is anders dan bij `document.getSelectionRect()` waar de rechthoek relatief is ten opzichte van het object dat wordt bewerkt.

Deze methode is hetzelfde als het slepen van een rechthoek met het gereedschap Selectie. Een instantie moet volledig door de rechthoek zijn ingesloten om te worden geselecteerd.

Als u een waarde voor *bContactSensitiveSelection* doorgeeft, is deze alleen geldig voor deze methode en heeft deze geen invloed op de selectiemodus Contactgevoelig voor het document (zie [fl.contactSensitiveSelection](#)

**Opmerking:** Als u `setSelectionRect()` herhaalt via het deelvenster of de menuopdracht Historie, wordt de stap herhaald die vóór de bewerking `setSelectionRect()` is uitgevoerd.

## Voorbeeld

In het volgende voorbeeld wordt de eerste selectie vervangen door de tweede selectie:

```
fl.getDocumentDOM().setSelectionRect({left:1, top:1, right:200, bottom:200});  
fl.getDocumentDOM().setSelectionRect({left:364.0, top:203.0, right:508.0, bottom:434.0},  
true);
```

In het volgende voorbeeld wordt de tweede selectie toegevoegd aan de eerste selectie: Dit is hetzelfde als het indrukken van Shift tijdens het selecteren van een tweede object.

```
fl.getDocumentDOM().setSelectionRect({left:1, top:1, right:200, bottom:200});  
fl.getDocumentDOM().setSelectionRect({left:364.0, top:203.0, right:508.0, bottom:434.0},  
false);
```

## Zie ook

[document.getSelectionRect\(\)](#), [document.selection](#), [document.setSelectionBounds\(\)](#)

## document.setStageVanishingPoint()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.setStageVanishingPoint(point)
```

### Parameters

**point** Een punt waarmee de *x*- en *y*-coördinaat worden opgegeven van de locatie waar het verdwijnpunt voor het weergeven van 3D-objecten moet worden ingesteld.

### Geretourneerde waarde

Niets.

### Beschrijving

Hiermee geeft u het verdwijnpunt voor het weergeven van 3D-objecten op.

### Voorbeeld

In het volgende voorbeeld wordt het verdwijnpunt van het werkgebied ingesteld:

```
fl.getDocumentDOM().setStageVanishingPoint({x:45, y:45});
```

## document.setStageViewAngle()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.setStageViewAngle(angle)
```

### Parameters

**angle** Een drijvende-kommawaarde tussen 0,0 en 179,0.

### Geretourneerde waarde

Niets.

### Beschrijving

Hiermee geeft u de perspectiefhoek voor het weergeven van 3D-objecten op.

### Voorbeeld

In het volgende voorbeeld wordt de perspectiefhoek van het werkgebied ingesteld op 70 graden:

```
fl.getDocumentDOM().setStageViewAngle(70);
```

## document.setStroke()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setStroke(color, size, strokeType)
```

### Parameters

**color** De kleur van de streek, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

**size** Een drijvende komma-waarde waarmee de nieuwe streekgrootte voor de selectie wordt opgegeven.

**strokeType** Een tekenreeks waarmee het nieuwe streektipe voor de selectie wordt opgegeven. Acceptabele waarden zijn "hairline", "solid", "dashed", "dotted", "ragged", "stipple" en "hatched".

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden de kleur, breedte en stijl van de geselecteerde streek ingesteld. Zie [document.setCustomStroke\(\)](#) voor informatie over het wijzigen van de streek in het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In het volgende voorbeeld wordt de kleur van de streek ingesteld op rood, de breedte op 3,25 en het type op gestreept:

```
fl.getDocumentDOM().setStroke("#ff0000", 3.25, "dashed");
```

## document.setStrokeColor()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setStrokeColor(color)
```

### Parameters

**color** De kleur van de streek, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de strekkleur van de selectie gewijzigd in de opgegeven kleur. Zie [document.setCustomStroke\(\)](#) voor informatie over het wijzigen van de streek in het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In de drie instructies in het volgende voorbeeld wordt de strekkleur ingesteld met elk van de verschillende notaties voor het opgeven van kleur.

```
fl.getDocumentDOM().setStrokeColor("#cc00cc");  
fl.getDocumentDOM().setStrokeColor(0xcc00cc);  
fl.getDocumentDOM().setStrokeColor(120000);
```

## document.setStrokeSize()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setStrokeSize(size)
```

### Parameters

**size** Een drijvende komma-waarde in het bereik van 0,25 tot en met 10 die de streekgrootte aangeeft. Bij deze methode wordt een precisie van meer dan twee decimalen genegeerd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de streekgrootte van de selectie gewijzigd in de opgegeven grootte. Zie [document.setCustomStroke\(\)](#) voor informatie over het wijzigen van de streek in het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In het volgende voorbeeld wordt de streekgrootte voor de selectie gewijzigd in 5:

```
fl.getDocumentDOM().setStrokeSize(5);
```

## document.setStrokeStyle()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setStrokeStyle(strokeType)
```

### Parameters

**strokeType** Een tekenreeks die de streekstijl voor de huidige selectie bevat. Acceptabele waarden zijn "hairline", "solid", "dashed", "dotted", "ragged", "stipple" en "hatched".

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de streekstijl van de selectie gewijzigd in de opgegeven stijl. Zie [document.setCustomStroke\(\)](#) voor informatie over het wijzigen van de streek in het deelvenster Gereedschappen en Eigenschapcontrole.

### Voorbeeld

In het volgende voorbeeld wordt de streekstijl voor de selectie gewijzigd in "dashed":

```
fl.getDocumentDOM().setStrokeStyle("dashed");
```

## document.setTextRectangle()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setTextRectangle(boundingRectangle)
```

### Parameters

**boundingRectangle** Een rechthoek die de nieuwe grootte aangeeft waarbinnen het tekstitem moet worden weergegeven. Zie [document.addNewRectangle\(\)](#) voor meer informatie over de notatie van *boundingRectangle*.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de grootte van ten minste één tekstveld is gewijzigd, anders `false`.

### Beschrijving

Methode; hiermee wordt het kader voor het geselecteerde tekstitem gewijzigd in de opgegeven grootte. Bij deze methode wordt de tekst opnieuw weergegeven binnen de nieuwe rechthoek; het tekstitem wordt niet geschaald of getransformeerd. De waarden die in *boundingRectangle* worden doorgegeven, worden als volgt gebruikt:

- Als de tekst horizontaal en statisch is, wordt alleen de breedte gebruikt die in *boundingRectangle* wordt doorgegeven; de hoogte wordt automatisch berekend zodat alle tekst in het kader past.
- Als de tekst verticaal (en dus statisch) is, wordt alleen de hoogte gebruikt die in *boundingRectangle* wordt doorgegeven; de breedte wordt automatisch berekend zodat alle tekst in het kader past.

- Bij dynamische tekst of invoertekst worden de breedte en de hoogte gebruikt die in *boundingRectangle* worden doorgegeven, en kan de resulterende rechthoek groter zijn dan nodig is voor alle tekst. Als met de parameters echter een rechthoek wordt opgegeven die te klein is voor alle tekst, wordt alleen de breedte gebruikt die in *boundingRectangle* wordt doorgegeven (de hoogte wordt automatisch berekend zodat alle tekst in het kader past).

### Voorbeeld

In het volgende voorbeeld wordt de grootte van het selectiekader voor de tekst gewijzigd in de opgegeven afmetingen:

```
fl.getDocumentDOM().setTextRectangle({left:0, top:0, right:50, bottom:200})
```

## document.setTextSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setTextSelection(startIndex, endIndex)
```

### Parameters

**startIndex** Een geheel getal dat de positie aangeeft van het eerste teken dat moet worden geselecteerd. De positie van het eerste teken is 0 (nul).

**endIndex** Een geheel getal dat de eindpositie van de selectie tot aan, maar niet inclusief, *endIndex* aangeeft. De positie van het eerste teken is 0 (nul).

### Geretoureerde waarde

Een Booleaanse waarde: `true` als de tekst kan worden geselecteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt in het huidige tekstveld de tekst geselecteerd die wordt bepaald door de waarden van *startIndex* en *endIndex*. De tekstbewerkingsmodus wordt geactiveerd als dat nog niet is gebeurd.

### Voorbeeld

In het volgende voorbeeld wordt de tekst van het 6e teken tot het 25e teken geselecteerd:

```
fl.document.setTextSelection(5, 25);
```

## document.setTextString()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setTextString(text [, startIndex [, endIndex]])
```



### Parameters

**text** Een tekenreeks met de tekens die moeten worden ingevoegd in het tekstveld.

**startIndex** Een geheel getal dat het eerste teken aangeeft dat moet worden vervangen. De positie van het eerste teken is 0 (nul). Deze parameter is optioneel.

**endIndex** Een geheel getal dat het laatste teken aangeeft dat moet worden vervangen. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de tekst van ten minste één tekenreeks is ingesteld, anders `false`.

### Beschrijving

Methode; hiermee wordt een tekenreeks ingevoegd. Als de optionele parameters niet worden doorgegeven, wordt de geselecteerde tekst vervangen; als het tekstobject momenteel niet wordt bewerkt, wordt de hele tekst vervangen. Als alleen *startIndex* wordt doorgegeven, wordt de tekenreeks op die positie ingevoegd. Als *startIndex* en *endIndex* worden doorgegeven, vervangt de tekenreeks het tekstsegment van *startIndex* tot aan, maar niet inclusief, *endIndex*.

### Voorbeeld

In het volgende voorbeeld wordt de huidige tekstselectie vervangen door “Hello World”:

```
var success = fl.getDocumentDOM().setTextString("Hello World!");
```

In het volgende voorbeeld wordt “hello” ingevoegd op positie 6 van de huidige tekstselectie:

```
var pass = fl.getDocumentDOM().setTextString("hello", 6);
```

In het volgende voorbeeld wordt “Howdy” ingevoegd van positie 2 tot aan, maar niet inclusief, positie 7 van de huidige tekstselectie:

```
var ok = fl.getDocumentDOM().setTextString("Howdy", 2, 7);
```

### Zie ook

[document.getTextString\(\)](#)

## document.setTransformationPoint()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.setTransformationPoint( transformationPoint )
```

### Parameters

**transformationPoint** Een punt (bijvoorbeeld {x:10, y:20}, waarbij x en y drijvende komma-waarden zijn) dat de waarden bevat van het transformatiepunt van elk van de volgende elementen:

- Vormen: *transformationPoint* wordt ten opzichte van het document ingesteld (0,0 is de linkerbovenhoek van het werkgebied).
- Symbolen: *transformationPoint* wordt ten opzichte van het registratiepunt van het symbool ingesteld (0,0 is het registratiepunt).

- Tekst: *transformationPoint* wordt ten opzichte van het tekstveld ingesteld (0,0 is de linkerbovenhoek van het tekstveld).
- Bitmaps/video's: *transformationPoint* wordt ingesteld ten opzichte van de bitmap/video (0,0 is de linkerbovenhoek van de bitmap of video).
- Tekenobjecten, primitieve ovaal en rechthoeken en groepen: *transformationPoint* wordt ten opzichte van het document ingesteld (0,0 is de linkerbovenhoek van het werkgebied). Als u het *transformatiepunt* ten opzichte van het middelpunt van het object, de primitieve vorm of de groep wilt instellen, gebruikt u `element.setTransformationPoint()`.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de positie van het transformatiepunt van de huidige selectie ingesteld.

#### Voorbeeld

In het volgende voorbeeld wordt het transformatiepunt van de huidige selectie ingesteld op 100, 200:

```
fl.getDocumentDOM().setTransformationPoint({x:100, y:200});
```

#### Zie ook

`document.getTransformationPoint()`, `element.setTransformationPoint()`

## document.silent

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.silent
```

#### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of het object toegankelijk is. Dit is gelijk aan de inverse logica van de instelling Film toegankelijk maken in het deelvenster Toegankelijkheid. Dit betekent dat wanneer `document.silent` de waarde `true` heeft, dit gelijk is aan het uitschakelen van de optie Film toegankelijk maken. Als de waarde `false` is, is dit gelijk aan het inschakelen van de optie Film toegankelijk maken.

#### Voorbeeld

In het volgende voorbeeld wordt de variabele `isSilent` ingesteld op de waarde van de eigenschap `silent`.

```
var isSilent = fl.getDocumentDOM().silent;
```

In het volgende voorbeeld wordt de eigenschap `silent` ingesteld op `false`. Dit geeft aan dat het document toegankelijk is:

```
fl.getDocumentDOM().silent = false;
```

## document.skewSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.skewSelection(xSkew, ySkew [, whichEdge])
```

### Parameters

**xSkew** Een drijvende komma-waarde waarmee de waarde van  $x$  in graden wordt opgegeven. Dit is de waarde waarmee moet worden scheefgetrokken.

**ySkew** Een drijvende komma-waarde waarmee de waarde van  $y$  in graden wordt opgegeven. Dit is de waarde waarmee moet worden scheefgetrokken.

**whichEdge** Een tekenreeks die de rand aangeeft waar de transformatie optreedt. Als u geen waarde opgeeft, wordt scheefgetrokken ten opzichte van het transformatiepunt. Acceptabele waarden zijn "top center", "right center", "bottom center" en "left center". Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de selectie een bepaald aantal graden scheefgetrokken. Het effect is hetzelfde als het scheeftrekken van het object met het gereedschap Vrije transformatie.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde object 2,0 graden verticaal en 1,5 graden horizontaal scheefgetrokken. In het tweede voorbeeld wordt het object getransformeerd ten opzichte van het midden van de bovenrand:

```
fl.getDocumentDOM().skewSelection(2.0, 1.5);  
fl.getDocumentDOM().skewSelection(2.0, 1.5, "top center");
```

## document.smoothSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.smoothSelection()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de curve van elke geselecteerde contourvulling of gekromde lijn vloeiend gemaakt. Deze methode voert dezelfde actie uit als de knop Vloeiend maken in het deelvenster Gereedschappen.

### Voorbeeld

In het volgende voorbeeld wordt de curve van de huidige selectie vloeiend gemaakt:

```
fl.getDocumentDOM().smoothSelection();
```

## document.sourcePath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.sourcePath
```

### Beschrijving

Eigenschap; een tekenreeks die een lijst bevat met items in het ActionScript 3.0-bronpad van het document, dat de locatie aangeeft van ActionScript-klassebestanden. Items in de tekenreeks worden gescheiden door puntkomma's. In het ontwerpgereedschap worden de items opgegeven door Bestand > Publicatie-instellingen te kiezen en vervolgens ActionScript 3.0-instellingen te kiezen op het tabblad Flash.

### Voorbeeld

In het volgende voorbeeld wordt de map ./Class files toegevoegd aan het bronpad van het document:

```
var myDoc = fl.getDocumentDOM();  
fl.trace(myDoc.sourcePath);  
myDoc.sourcePath = "./Class files;" + myDoc.sourcePath;  
fl.trace(myDoc.sourcePath);
```

### Zie ook

[document.externalLibraryPath](#), [document.libraryPath](#), [fl.sourcePath](#)

## document.space()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.space(direction [, bUseDocumentBounds])
```

### Parameters

**direction** Een tekenreeks die de richting aangeeft waarin de objecten in de selectie worden verdeeld. Acceptabele waarden zijn "horizontal" en "vertical".

**bUseDocumentBounds** Een Booleaanse waarde: als de waarde is ingesteld op `true`, worden de objecten langs de documentgrenzen verdeeld. Anders worden de grenzen van de geselecteerde objecten gebruikt. De standaardwaarde is `false`. Deze parameter is optioneel.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee worden de objecten in de selectie gelijkmatig verdeeld.

#### Voorbeeld

In het volgende voorbeeld worden de objecten horizontaal ten opzichte van het werkgebied verdeeld:

```
fl.getDocumentDOM().space("horizontal", true);
```

In het volgende voorbeeld worden de objecten horizontaal ten opzichte van elkaar verdeeld:

```
fl.getDocumentDOM().space("horizontal");
```

In het volgende voorbeeld worden de objecten horizontaal ten opzichte van elkaar verdeeld, terwijl *bUseDocumentBounds* expliciet is ingesteld op `false`:

```
fl.getDocumentDOM().space("horizontal", false);
```

#### Zie ook

[document.getAlignToDocument\(\)](#), [document.setAlignToDocument\(\)](#)

## document.straightenSelection()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
document.straightenSelection()
```

#### Parameters

Geen.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee worden de geselecteerde streken rechtgetrokken. Deze methode voert dezelfde bewerking uit als de knop *Rechttrekken* in het deelvenster *Gereedschappen*.

#### Voorbeeld

In het volgende voorbeeld wordt de curve van de huidige selectie rechtgetrokken:

```
fl.getDocumentDOM().straightenSelection();
```

## document.swapElement()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.swapElement (name)
```

### Parameters

**name** Een tekenreeks met de naam van het bibliotheekitem dat moet worden gebruikt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de huidige selectie omgewisseld met de opgegeven selectie. De selectie moet een grafische afbeelding, knop, filmclip, video of bitmap bevatten. Er wordt een foutbericht weergegeven als er geen object is geselecteerd of als het opgegeven object niet is gevonden.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie omgewisseld met `Symbol 1` uit de bibliotheek:

```
fl.getDocumentDOM().swapElement('Symbol 1');
```

## document.swapStrokeAndFill()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.swapStrokeAndFill()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden streek- en vulkleur omgewisseld.

### Voorbeeld

In het volgende voorbeeld worden de streek- en vulkleur in het huidige document omgewisseld:

```
fl.getDocumentDOM().swapStrokeAndFill();
```

## document.testMovie()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.testMovie([Boolean abortIfErrorsExist])
```

### Parameters

**abortIfErrorsExist** Booleaanse waarde, de standaardinstelling is false. Als deze waarde wordt ingesteld op true, wordt de testsessie voor de film niet gestart en wordt het SWF-venster niet geopend in geval van compilatiefouten. Compilatiawaarschuwingen leiden niet tot het afbreken van de opdracht. Deze parameter is toegevoegd in Flash Professional CS5.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de bewerking Film testen uitgevoerd op het document.

### Voorbeeld

In het volgende voorbeeld wordt de film voor het huidige document getest, maar wordt de testfilm afgebroken in geval van compilatiefouten:

```
fl.getDocumentDOM().testMovie(1);
```

### Zie ook

[document.canTestMovie\(\)](#), [document.testScene\(\)](#)

## document.testScene()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.testScene()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de bewerking Scène testen uitgevoerd op de huidige scène van het document.

### Voorbeeld

In het volgende voorbeeld wordt de huidige scène in het document getest:

```
fl.getDocumentDOM().testScene();
```

### Zie ook

[document.canTestScene\(\)](#), [document.testMovie\(\)](#)

## document.timelines

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.timelines`

### Beschrijving

Alleen-lezen eigenschap; een array met tijdlijnobjecten (zie [Object Timeline](#)).

### Voorbeeld

In het volgende voorbeeld wordt de array met huidige tijdlijnen in het actieve document opgehaald en worden de namen weergegeven in het deelvenster Uitvoer:

```
var i = 0;
var curTimelines = fl.getDocumentDOM().timelines;
while(i < fl.getDocumentDOM().timelines.length){
    alert(curTimelines[i].name);
    ++i;
}
```

### Zie ook

[document.currentTimeline](#), [document.getTimeline\(\)](#)

## document.traceBitmap()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`document.traceBitmap(threshold, minimumArea, curveFit, cornerThreshold)`

### Parameters

**threshold** Een geheel getal dat het aantal kleuren in uw overgetrokken bitmap bepaalt. Acceptabele waarden zijn gehele getallen in het bereik van 0 tot en met 500.

**minimumArea** Een geheel getal dat de straal in pixels aangeeft. Acceptabele waarden zijn gehele getallen in het bereik van 1 tot en met 1000.



**curveFit** Een tekenreeks die aangeeft hoe vloeiend omtrekken worden getekend. Acceptabele waarden zijn "pixels", "very tight", "tight", "normal", "smooth" en "very smooth".

**cornerThreshold** Een soortgelijke tekenreeks als *curveFit*, maar heeft betrekking op de hoeken van de bitmapafbeelding. Acceptabele waarden zijn "many corners", "normal" en "few corners".

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de huidige selectie als bitmap overgetrokken. Deze methode is hetzelfde als het selecteren van Wijzigen > Bitmap > Bitmap overtrekken.

#### Voorbeeld

In het volgende voorbeeld wordt de geselecteerde bitmap overgetrokken met de opgegeven parameters:

```
fl.getDocumentDOM().traceBitmap(0, 500, 'normal', 'normal');
```

## document.translate3DCenter()

#### Beschikbaarheid

Flash CS4 Professional.

#### Gebruik

```
document.translate3DCenter(xyzCoordinate)
```

#### Parameters

**xyzCoordinate** Een XYZ-coördinaat die het middelpunt voor 3D-rotatie of -omzetting aangeeft.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode: hiermee de XYZ-positie ingesteld ten opzichte waarvan de selectie wordt omgezet of geroteerd. Deze methode is alleen beschikbaar voor filmclips.

#### Voorbeeld

In het volgende voorbeeld worden de XYZ-assen voor 3D-omzetting opgegeven:

```
fl.getDocumentDOM().translate3DCenter({x:180, y:18, z:-30});
```

## document.translate3DSelection()

#### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
document.translate3DSelection(xyzCoordinate, bGlobalTransform)
```

### Parameters

**xyzCoordinate** Een XYZ-coördinaat die de assen voor 3D-omzetting aangeeft.

**bGlobalTransform** Een Booleaanse waarde die opgeeft of de transformatiemodus globaal (`true`) of lokaal (`false`) is.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode: hiermee wordt een 3D-omzetting toegepast op de selectie. Deze methode is alleen beschikbaar voor filmclips.

### Voorbeeld

In het volgende voorbeeld wordt de selectie eerst omgezet ten opzichte van het werkgebied (globaal) en vervolgens ten opzichte van zichzelf (lokaal).

```
var myDocument = fl.getDocumentDOM();  
myDocument.translate3DSelection({x:52.0, y:0, z:0}, true);  
myDocument.translate3DSelection({x:52.0, y:0, z:-55.2}, false);
```

### Zie ook

[document.translate3DCenter\(\)](#)

## document.transformSelection()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.transformSelection(a, b, c, d)
```

### Parameters

**a** Een drijvende komma-waarde waarmee het element (0,0) van de transformatiematrix wordt opgegeven.

**b** Een drijvende komma-waarde waarmee het element (0,1) van de transformatiematrix wordt opgegeven.

**c** Een drijvende komma-waarde waarmee het element (1,0) van de transformatiematrix wordt opgegeven.

**d** Een drijvende komma-waarde waarmee het element (1,1) van de transformatiematrix wordt opgegeven.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een algemene transformatie op de huidige selectie uitgevoerd door het toepassen van de matrix die in de argumenten is opgegeven. Zie de eigenschap `element.matrix` voor meer informatie.

### Voorbeeld

In het volgende voorbeeld wordt de selectie uitgerekt met een factor 2 in de x-richting:

```
fl.getDocumentDOM().transformSelection(2.0, 0.0, 0.0, 1.0);
```

## document.unGroup()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.unGroup()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de groepering van de huidige selectie opgeheven.

### Voorbeeld

In het volgende voorbeeld wordt de groepering van de elementen in de huidige selectie opgeheven:

```
fl.getDocumentDOM().unGroup();
```

### Zie ook

[document.group\(\)](#)

## document.union()

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.union()
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol was, anders `false`.

### Beschrijving

Methode; hiermee worden alle geselecteerde vormen gecombineerd tot een tekenobject.

### Voorbeeld

In het volgende voorbeeld worden alle geselecteerde vormen gecombineerd tot een tekenobject:

```
fl.getDocumentDOM().union();
```

### Zie ook

[document.crop\(\)](#), [document.deleteEnvelope\(\)](#), [document.intersect\(\)](#), [document.punch\(\)](#),  
[shape.isDrawingObject](#)

## document.unlockAllElements()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.unlockAllElements();
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle vergrendelde elementen in het geselecteerde frame ontgrendeld.

### Voorbeeld

In het volgende voorbeeld worden alle vergrendelde objecten in het huidige frame ontgrendeld:

```
fl.getDocumentDOM().unlockAllElements();
```

### Zie ook

[element.locked](#)

## document.viewMatrix

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.viewMatrix
```

### Beschrijving

Alleen-lezen eigenschap; een object `Matrix.viewMatrix` wordt gebruikt voor het omzetten van de objectruimte naar de documentruimte wanneer het document wordt bewerkt. Wanneer een gereedschap de locatie van de muis ontvangt, is deze relatief ten opzichte van het object dat momenteel wordt bewerkt. Zie [Object Matrix](#).

Als u bijvoorbeeld een symbool maakt, erop dubbelklikt om het te bewerken en tekent met het gereedschap Meerpuntige ster, is het punt (0,0) het registratiepunt van het symbool. Het object `drawingLayer` verwacht echter waarden in de documentruimte. Als u vanaf (0,0) een lijn tekent met behulp van `drawingLayer`, begint deze daarom in de linkerbovenhoek van het werkgebied. Met de eigenschap `viewMatrix` kunt u de ruimte van het object dat wordt bewerkt, omzetten naar de documentruimte.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `viewMatrix` opgehaald:

```
var mat = fl.getDocumentDOM().viewMatrix;
```

## document.width

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.width
```

### Beschrijving

Eigenschap; een geheel getal dat de breedte van het document (werkgebied) in pixels aangeeft.

### Voorbeeld

In het volgende voorbeeld wordt de breedte van het werkgebied ingesteld op 400 pixels:

```
fl.getDocumentDOM().width= 400;
```

### Zie ook

[document.height](#)

## document.xmlPanel()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
document.xmlPanel (fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad bevat van het XML-bestand waarin de besturingselementen in het deelvenster zijn gedefinieerd. U moet het volledige pad opgeven.

### Geretourneerde waarde

Een object waarin eigenschappen zijn gedefinieerd voor alle besturingselementen die in het XML-bestand zijn gedefinieerd. Alle eigenschappen worden geretourneerd als tekenreeksen. Het geretourneerde object bevat een vooraf gedefinieerde eigenschap met de naam "dismiss" die de waarde "accept" of "cancel" heeft.

### Beschrijving

Methode; hiermee wordt een XMLUI-dialoogvenster gepost. Zie [fl.xmlui](#).

### Voorbeeld

In het volgende voorbeeld wordt het bestand Test.xml geladen en worden alle eigenschappen weergegeven die zich daarin bevinden:

```
var obj = fl.getDocumentDOM().xmlPanel(fl.configURI + "Commands/Test.xml");
for (var prop in obj) {
    fl.trace("property " + prop + " = " + obj[prop]);
}
```

## document.zoomFactor

### Beschikbaarheid

Flash 8.

### Gebruik

```
document.zoomFactor
```

### Beschrijving

Eigenschap; hiermee wordt het zoompercentage van het werkgebied tijdens het ontwerpen bepaald. De waarde 1 betekent 100 procent zoomen, 8 betekent 800 procent, 0,5 betekent 50 procent, enzovoort.

### Voorbeeld

In het volgende voorbeeld wordt de zoomfactor van het werkgebied ingesteld op 200 procent:

```
fl.getDocumentDOM().zoomFactor = 2;
```

# Hoofdstuk 12: Object drawingLayer

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object drawingLayer is toegankelijk vanuit JavaScript als onderliggend element van het object flash. Het object drawingLayer wordt gebruikt voor uitbreidbare gereedschappen wanneer de gebruiker tijdens het slepen tijdelijk wil tekenen, bijvoorbeeld bij het maken van een selectiekader. U moet `drawingLayer.beginFrame()` aanroepen voordat u andere methoden drawingLayer aanroept.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object drawingLayer:

Methode	Beschrijving
<code>drawingLayer.beginDraw()</code>	Hiermee wordt de tekenmodus in Flash geactiveerd.
<code>drawingLayer.beginFrame()</code>	Hiermee wordt gewist wat eerder werd getekend met de drawingLayer en worden andere tekenopdrachten voorbereid.
<code>drawingLayer.cubicCurveTo()</code>	Tekent een kubische curve vanaf de huidige penlocatie waarbij de parameters worden gebruikt als de coördinaten van het kubische segment.
<code>drawingLayer.curveTo()</code>	Hiermee wordt een kwadratisch curvesegment getekend dat begint op de huidige tekenpositie en eindigt op een opgegeven punt.
<code>drawingLayer.drawPath()</code>	Hiermee wordt het opgegeven pad getekend.
<code>drawingLayer.endDraw()</code>	Hiermee wordt de tekenmodus afgesloten.
<code>drawingLayer.endFrame()</code>	Hiermee wordt het einde van een groep tekenopdrachten gesignaleerd.
<code>drawingLayer.lineTo()</code>	Hiermee wordt een lijn getekend vanaf de huidige tekenpositie naar het punt (x,y).
<code>drawingLayer.moveTo()</code>	Hiermee wordt de huidige tekenpositie ingesteld.
<code>drawingLayer.newPath()</code>	Hiermee wordt een nieuw <a href="#">Object Path</a> geretourneerd.
<code>drawingLayer.setColor()</code>	Hiermee wordt de kleur ingesteld van volgende tekengegevens.
<code>drawingLayer.setFill()</code>	Deze methode is niet beschikbaar.
<code>drawingLayer.setStroke()</code>	Deze methode is niet beschikbaar.

## drawingLayer.beginDraw()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.beginDraw([persistentDraw])
```

### Parameters

**`persistentDraw`** Een Booleaanse waarde (optioneel). Als de waarde is ingesteld op `true`, blijft de tekening in het laatste frame in het werkgebied totdat `beginDraw()` of `beginFrame()` opnieuw wordt aangeroepen. (In deze context verwijst *frame* naar waar u begint en ophoudt met tekenen, de term verwijst niet naar frames in de tijdlijn.) Wanneer gebruikers bijvoorbeeld een rechthoek tekenen, kunnen ze een voorbeeld van de contour van de vorm bekijken terwijl ze slepen met de muis. Als u wilt dat de voorbeeldvorm behouden blijft wanneer de gebruiker de muisknop loslaat, stelt u *`persistentDraw`* in op `true`.

### Geretoureerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de tekenmodus in Flash geactiveerd. De tekenmodus wordt gebruikt om tijdelijk te tekenen terwijl de muisknop is ingedrukt. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

### Voorbeeld

In het volgende voorbeeld wordt de tekenmodus in Flash geactiveerd:

```
fl.drawingLayer.beginDraw();
```

## `drawingLayer.beginFrame()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.beginFrame()
```

### Parameters

Geen.

### Geretoureerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt gewist wat eerder werd getekend met de `drawingLayer` en worden meer tekenopdrachten voorbereid. Moet worden aangeroepen na `drawingLayer.beginDraw()`. Alles wat is getekend tussen `drawingLayer.beginFrame()` en `drawingLayer.endFrame()` blijft in het werkgebied totdat u `beginFrame()` en `endFrame()` opnieuw aanroept. (In deze context verwijst *frame* naar waar u begint en ophoudt met tekenen, de term verwijst niet naar frames in de tijdlijn.) Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen. Zie [`drawingLayer.beginDraw\(\)`](#).



## `drawingLayer.cubicCurveTo()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.cubicCurveTo(x1Ctrl, y1Ctrl, x2Ctl, y2Ctl, xEnd, yEnd)
```

### Parameters

**x1Ctl** Een drijvende-kommawaarde die de  $x$ -locatie is van het eerste besturingspunt.

**y1Ctl** Een drijvende-kommawaarde die de  $y$ -locatie is van het eerste besturingspunt.

**x2Ctl** Een drijvende-kommawaarde die de  $x$ -positie is van het middelste besturingspunt.

**y2Ctl** Een drijvende-kommawaarde die de  $y$ -positie is van het middelste besturingspunt.

**xEnd** Een drijvende-kommawaarde die de  $x$ -positie is van het laatste besturingspunt.

**yEnd** Een drijvende-kommawaarde die de  $y$ -positie is van het laatste besturingspunt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een cubische curve getekend vanaf de huidige penlocatie waarbij de parameters worden gebruikt als de coördinaten van het cubische segment. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

### Voorbeeld

In het volgende voorbeeld wordt een cubische curve getekend met behulp van de opgegeven besturingspunten:

```
fl.drawingLayer.cubicCurveTo(0, 0, 1, 1, 2, 0);
```

## `drawingLayer.curveTo()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.curveTo(xCtl, yCtl, xEnd, yEnd)
```

### Parameters

**xCtl** Een drijvende-kommawaarde die de  $x$ -positie is van het besturingspunt.

**yCtl** Een drijvende-kommawaarde die de  $y$ -positie is van het besturingspunt.

**xEnd** Een drijvende-kommawaarde die de  $x$ -positie is van het laatste besturingspunt.

**yEnd** Een drijvende-kommawaarde die de  $y$ -positie is van het laatste besturingspunt.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een kwadratisch curvesegment getekend dat begint op de huidige tekenpositie en eindigt op een opgegeven punt. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

#### Voorbeeld

In het volgende voorbeeld wordt een kwadratische curve getekend met behulp van de opgegeven besturingspunten:

```
fl.drawingLayer.curveTo(0, 0, 2, 0);
```

## `drawingLayer.drawPath()`

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
drawingLayer.drawPath(path)
```

#### Parameters

**path** Een [Object Path](#) dat moet worden getekend.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt het pad getekend dat wordt opgegeven door de parameter *path*. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

#### Voorbeeld

In het volgende voorbeeld wordt een pad getekend dat wordt opgegeven door het object Path met de naam `gamePath`:

```
fl.drawingLayer.drawPath(gamePath);
```

## `drawingLayer.endDraw()`

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
drawingLayer.endDraw()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de tekenmodus verlaten. De tekenmodus wordt gebruikt om tijdelijk te tekenen terwijl de muisknop is ingedrukt. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

### Voorbeeld

In het volgende voorbeeld wordt de tekenmodus verlaten:

```
fl.drawingLayer.endDraw();
```

## `drawingLayer.endFrame()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.endFrame();
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het einde van een groep tekenopdrachten gesignaleerd. Een groep tekenopdrachten verwijst naar alles wat is getekend tussen `drawingLayer.beginFrame()` en `drawingLayer.endFrame()`. Bij de volgende aanroep van `drawingLayer.beginFrame()` wordt alles gewist wat werd getekend in deze groep tekenopdrachten. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

## `drawingLayer.lineTo()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.lineTo(x, y)
```

#### Parameters

**x** Een drijvende-kommawaarde die de  $x$ -coördinaat is van het eindpunt van de lijn die moet worden getekend.

**y** Een drijvende-kommawaarde die de  $y$ -coördinaat is van het eindpunt van de lijn die moet worden getekend.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een lijn getekend vanaf de huidige tekenpositie naar het punt  $(x,y)$ . Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

#### Voorbeeld

In het volgende voorbeeld een lijn getekend vanaf de huidige tekenpositie naar het punt (20,30):

```
fl.drawingLayer.lineTo(20, 30);
```

## `drawingLayer.moveTo()`

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
drawingLayer.moveTo(x, y)
```

#### Parameters

**x** Een drijvende-kommawaarde die de  $x$ -coördinaat opgeeft van de positie waarop het tekenen moet beginnen.

**y** Een drijvende-kommawaarde die de  $y$ -coördinaat opgeeft van de positie waarop het tekenen moet beginnen.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de huidige tekenpositie ingesteld. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen.

#### Voorbeeld

In het volgende voorbeeld wordt de huidige tekenpositie ingesteld op het punt (10,15):

```
fl.drawingLayer.moveTo(10, 15);
```

## `drawingLayer.newPath()`

#### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.newPath()
```

### Parameters

Geen.

### Geretourneerde waarde

Een object `Path`.

### Beschrijving

Methode, hiermee wordt een nieuw object `Path` geretourneerd. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen. Zie [Object Path](#).

### Voorbeeld

In het volgende voorbeeld wordt een nieuw object `Path` geretourneerd.

```
fl.drawingLayer.newPath();
```

## `drawingLayer.setColor()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
drawingLayer.setColor(color)
```

### Parameters

`color` De kleur van gegevens die erna worden getekend, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de kleur ingesteld van gegevens die erna worden getekend. Is alleen van toepassing op permanente gegevens. Als u deze methode wilt gebruiken, moet de parameter die wordt doorgegeven aan `drawingLayer.beginDraw()`, zijn ingesteld op `true`. Normaalgesproken gebruikt u deze methode alleen voor het maken van uitbreidbare gereedschappen. Zie [drawingLayer.beginDraw\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt een rode lijn in het werkgebied getekend:

```
fl.drawingLayer.beginDraw( true );  
fl.drawingLayer.beginFrame();  
fl.drawingLayer.setColor( "#ff0000" );  
fl.drawingLayer.moveTo(0,0);  
fl.drawingLayer.lineTo(100,100);  
fl.drawingLayer.endFrame();  
fl.drawingLayer.endDraw();
```

## **`drawingLayer.setFill()`**

Deze methode is niet beschikbaar.

## **`drawingLayer.setStroke()`**

Deze methode is niet beschikbaar.

# Hoofdstuk 13: Object Edge

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Edge vertegenwoordigt een rand van een vorm in het werkgebied.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Edge:

Methode	Beschrijving
<code>edge.getControl()</code>	Hiermee wordt een puntobject opgehaald dat is ingesteld op de locatie van het opgegeven besturingspunt van de rand.
<code>edge.getHalfEdge()</code>	Hiermee wordt een object <a href="#">Object HalfEdge</a> geretourneerd.
<code>edge.setControl()</code>	Hiermee wordt de positie ingesteld van het besturingspunt van de rand.
<code>edge.splitEdge()</code>	Hiermee wordt de rand in twee delen gesplitst.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Edge:

Eigenschap	Beschrijving
<code>edge.cubicSegmentIndex</code>	Een geheel getal dat de indexwaarde van een kubisch segment van de rand opgeeft.
<code>edge.id</code>	Alleen-lezen; een geheel getal dat een unieke id voor de rand vertegenwoordigt.
<code>edge.isLine</code>	Alleen-lezen; een geheel getal met de waarde 0 of 1.
<code>edge.stroke</code>	Een <a href="#">Object Stroke</a> .

## edge.cubicSegmentIndex

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`edge.cubicSegmentIndex`

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat de indexwaarde van een cubisch segment van de rand opgeeft (zie [shape.getCubicSegmentPoints\(\)](#)).

### Voorbeeld

Met de volgende code worden de indexwaarden van alle kubische segmenten van de opgegeven rand weergegeven:

```
var theShape = fl.getDocumentDOM().selection[0];  
var edgesArray = theShape.edges;  
for (var i=0; i<edgesArray.length; i++) {  
    fl.trace(edgesArray[i].cubicSegmentIndex);  
}
```

## edge.getControl()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
edge.getControl(i)
```

### Parameters

**i** Een geheel getal dat opgeeft welk besturingspunt van de rand moet worden geretourneerd. Geef 0 op voor het eerste besturingspunt, 1 voor het middelste besturingspunt of 2 voor het laatste besturingspunt. Als de eigenschap [edge.isLine](#) `true` is, wordt het middelste besturingspunt ingesteld op het middelpunt van het segment dat het eerste en laatste besturingspunt verbindt.

### Geretourneerde waarde

Het opgegeven besturingspunt.

### Beschrijving

Methode; hiermee wordt een puntobject opgehaald dat is ingesteld op de locatie van het opgegeven besturingspunt van de rand.

### Voorbeeld

In het volgende voorbeeld wordt het eerste besturingspunt van de opgegeven vorm opgeslagen in de variabele `pt`:

```
var shape = fl.getDocumentDOM().selection[0];  
var pt = shape.edges[0].getControl(0);
```

## edge.getHalfEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
edge.getHalfEdge(index)
```

### Parameters

**index** Een geheel getal dat opgeeft welke halve rand moet worden geretourneerd. De waarde van *index* moet 0 zijn voor de eerste halve rand of 1 voor de tweede halve rand.



### Geretourneerde waarde

Een object HalfEdge.

### Beschrijving

Methode; retourneert een [Object HalfEdge](#).

### Voorbeeld

In het volgende voorbeeld worden de halve randen van de opgegeven rand opgeslagen in de variabelen `hEdge0` en `hEdge1`:

```
var shape = fl.getDocumentDOM().selection[0];  
var edge = shape.edges[0];  
var hEdge0 = edge.getHalfEdge(0);  
var hEdge1 = edge.getHalfEdge(1);
```

## edge.id

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
edge.id
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat een unieke id voor de rand vertegenwoordigt.

### Voorbeeld

In het volgende voorbeeld wordt een unieke id voor de opgegeven rand opgeslagen in de variabele `my_shape_id`:

```
var shape = fl.getDocumentDOM().selection[0];  
var my_shape_id = shape.edges[0].id;
```

## edge.isLine

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
edge.isLine
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal met de waarde 0 of 1. De waarde 1 geeft aan dat de rand een rechte lijn is. In dit geval doorsnijdt het middelste besturingspunt de lijn die de twee eindpunten verbindt.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of de opgegeven rand een rechte lijn is en wordt in het deelvenster Uitvoer de waarde 1 weergegeven als de rand een rechte lijn is, en 0 als de rand geen rechte lijn is:

```
var shape = fl.getDocumentDOM().selection[0];  
fl.trace(shape.edges[0].isLine);
```

## edge.setControl()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
edge.setControl(index, x, y)
```

### Parameters

**index** Een geheel getal dat opgeeft welk besturingspunt moet worden ingesteld. Gebruikt de waarden 0, 1 en 2 om respectievelijk het eerste, middelste en laatste besturingspunt op te geven.

**x** Een drijvende-kommawaarde die de horizontale locatie van het besturingspunt opgeeft. Als het werkgebied wordt weergegeven in de modus Bewerken of Op plaats bewerken, is de puntcoördinaat ten opzichte van het bewerkte object. Anders is de puntcoördinaat ten opzichte van het werkgebied.

**y** Een drijvende-kommawaarde die de verticale locatie van het besturingspunt opgeeft. Als het werkgebied wordt weergegeven in de modus Bewerken of Op plaats bewerken, is de puntcoördinaat ten opzichte van het bewerkte object. Anders is de puntcoördinaat ten opzichte van het werkgebied.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de positie ingesteld van het besturingspunt van de rand. U moet `shape.beginEdit()` aanroepen voordat u deze methode gebruikt. Zie [shape.beginEdit\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt het eerste besturingspunt van de opgegeven rand ingesteld op de coördinaten (0, 1):

```
x = 0; y = 1;  
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.edges[0].setControl(0, x, y);  
shape.endEdit();
```

## edge.splitEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
edge.splitEdge(t)
```

### Parameters

**t** Een drijvende-kommawaarde tussen 0 en 1 die opgeeft waar de rand moet worden gesplitst. Een waarde 0 vertegenwoordigt het ene eindpunt en een waarde 1 het andere. Als u bijvoorbeeld de waarde 0.5 doorgeeft, wordt de rand in het midden gesplitst, wat voor een lijn precies in het midden is. Als de rand een curve vertegenwoordigt, vertegenwoordigt 0.5 het parametrische midden van de curve.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de rand in twee delen gesplitst. U moet [shape.beginEdit\(\)](#) aanroepen voordat u deze methode gebruikt.

### Voorbeeld

In het volgende voorbeeld wordt de opgegeven rand in tweeën gesplitst:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit()  
shape.edges[0].splitEdge( 0.5 );  
shape.endEdit()
```

## edge.stroke

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
edge.stroke
```

### Beschrijving

Eigenschap: een [Object Stroke](#).

### Voorbeeld

Het volgende voorbeeld geeft de strekkleur weer van de eerste rand van het geselecteerde object:

```
var shape = fl.getDocumentDOM().selection[0];  
fl.trace(shape.edges[0].stroke.color);
```

# Hoofdstuk 14: Object Element

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Alles wat op het werkgebied wordt weergegeven, is van het type Element. Aan de hand van het volgende codevoorbeeld kunt u een element selecteren:

```
var el = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
```

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Element:

Methode	Beschrijving
<code>element.getPersistentData()</code>	Hiermee wordt de waarde opgehaald van de gegevens die zijn aangeduid met de parameter <i>name</i> .
<code>element.getTransformationPoint()</code>	Hiermee wordt de waarde opgehaald van het transformatiepunt van het opgegeven element.
<code>element.hasPersistentData()</code>	Hiermee wordt bepaald of de opgegeven gegevens zijn gekoppeld aan het opgegeven element.
<code>element.removePersistentData()</code>	Hiermee worden eventuele permanente gegevens met de opgegeven naam verwijderd die aan het object zijn gekoppeld.
<code>element.setPersistentData()</code>	Hiermee worden gegevens met een element opgeslagen.
<code>element.setTransformationPoint()</code>	Hiermee wordt de positie van het transformatiepunt van het opgegeven element ingesteld.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Element:

Eigenschap	Beschrijving
<code>element.depth</code>	Alleen-lezen; een geheel getal dat een waarde heeft die groter is dan 0 voor de diepte van het object in de weergave.
<code>element.elementType</code>	Alleen-lezen; een tekenreeks die staat voor het type van het opgegeven element.
<code>element.height</code>	Een drijvende-kommawaarde die de hoogte van het element in pixels aangeeft.
<code>element.layer</code>	Alleen-lezen; staat voor het <a href="#">Object Layer</a> waarop het element zich bevindt.
<code>element.left</code>	Alleen-lezen; een drijvende-kommawaarde die staat voor de linkerkant van het element.
<code>element.locked</code>	Een Booleaanse waarde: <code>true</code> als het element vergrendeld is, anders <code>false</code> .
<code>element.matrix</code>	Een <a href="#">Object Matrix</a> . Een matrix heeft de eigenschappen <code>a</code> , <code>b</code> , <code>c</code> , <code>d</code> , <code>tx</code> en <code>ty</code> . <code>a</code> , <code>b</code> , <code>c</code> en <code>d</code> zijn drijvende-kommawaarden; <code>tx</code> en <code>ty</code> zijn coördinaten.
<code>element.name</code>	Een tekenreeks die de naam van het element aanduidt. Wordt meestal de naam van de instantie genoemd.

Eigenschap	Beschrijving
<code>element.rotation</code>	Een geheel getal of drijvende-kommawaarde tussen -180 en 180 waarmee de draaiing (met de klok mee) van het object wordt aangegeven, in graden.
<code>element.scaleX</code>	Een drijvende-kommawaarde die de x-schaalwaarde van symbolen, tekenobjecten en primitieve rechthoeken en ovals aanduidt.
<code>element.scaleY</code>	Een drijvende-kommawaarde die de y-schaalwaarde van symbolen, tekenobjecten en primitieve rechthoeken en ovals aanduidt.
<code>element.selected</code>	Een Booleaanse waarde die aangeeft of het element is geselecteerd.
<code>element.skewX</code>	Een drijvende-kommawaarde tussen -180 en 180 die de x-waarde voor scheeftrekken van symbolen, tekenobjecten en primitieve rechthoeken en ovals aanduidt.
<code>element.skewY</code>	Een drijvende-kommawaarde tussen -180 en 180 die de y-waarde voor scheeftrekken van symbolen, tekenobjecten en primitieve rechthoeken en ovals aanduidt.
<code>element.top</code>	Alleen-lezen; bovenzijde van het element.
<code>element.transformX</code>	Een drijvende-kommawaarde die de x-waarde van het transformatiepunt van het geselecteerde element aanduidt, binnen het coördinatensysteem van het bovenliggende element van het element.
<code>element.transformY</code>	Een drijvende-kommawaarde die de y-waarde van het transformatiepunt van het geselecteerde element aanduidt, binnen het coördinatensysteem van het bovenliggende element van het element.
<code>element.width</code>	Een drijvende-kommawaarde die de breedte van het element in pixels aangeeft.
<code>element.x</code>	Een drijvende-kommawaarde die de x-waarde van het registratiepunt van het geselecteerde element aanduidt.
<code>element.y</code>	Een drijvende-kommawaarde die de y-waarde van het registratiepunt van het geselecteerde element aanduidt.

## element.depth

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`element.depth`

### Beschrijving

Alleen-lezen eigenschap; een geheel getal met een waarde groter dan 0 voor de diepte van het object in de weergave. De tekenvolgorde van objecten op het werkgebied duidt aan welk object boven de andere objecten ligt. De objectvolgorde kan ook worden ingesteld met de menuoptie Wijzigen > Rangschikken.

### Voorbeeld

In het volgende voorbeeld wordt de diepte van het opgegeven element weergegeven in het deelvenster Uitvoer:

```
// Select an object and run this script.  
fl.trace("Depth of selected object: " + fl.getDocumentDOM().selection[0].depth);
```

Zie het voorbeeld voor `element.elementType`.

## element.elementType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.elementType
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die staat voor het type van het opgegeven element. Geldige waarden zijn: "shape", "text", "instance" of "shapeObj". Een "shapeObj" wordt gemaakt met een uitbreidbaar gereedschap.

### Voorbeeld

In het volgende voorbeeld wordt het type van het eerste element opgeslagen in de variabele `eType`:

```
// In a new file, place a movie clip on first frame top layer, and  
// then run this line of script.  
var eType = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].elementType; //  
eType = instance
```

In het volgende voorbeeld worden verschillende eigenschappen weergegeven voor alle elementen in de huidige laag of frame:

```
var tl = fl.getDocumentDOM().getTimeline()  
var elts = tl.layers[tl.currentLayer].frames[tl.currentFrame].elements;  
for (var x = 0; x < elts.length; x++) {  
    var elt = elts[x];  
    fl.trace("Element "+ x +" Name = " + elt.name + " Type = " + elt.elementType + " location  
= " + elt.left + ", " + elt.top + " Depth = " + elt.depth);  
}
```

## element.getPersistentData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.getPersistentData(name)
```

### Parameters

**name** Een tekenreeks die de te retourneren gegevens identificeert.

### Geretourneerde waarde

De gegevens die worden opgegeven door de parameter *name*, of 0 als de gegevens niet bestaan.

### Beschrijving

Methode; hiermee wordt de waarde opgehaald van de gegevens die worden opgegeven door de parameter *name*. Het type gegevens hangt af van het type gegevens dat is opgeslagen (zie `element.setPersistentData()`). Alleen symbolen en bitmaps ondersteunen permanente gegevens.

### Voorbeeld

In het volgende voorbeeld worden gegevens voor het opgegeven element ingesteld en opgehaald, worden de waarden ervan in het deelvenster Uitvoer weergegeven en worden vervolgens de gegevens verwijderd:

```
// At least one symbol or bitmap is selected in the first layer, first frame.
var elt = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
elt.setPersistentData("myData","integer", 12);
if (elt.hasPersistentData("myData")){
fl.trace("myData = "+ elt.getPersistentData("myData"));
elt.removePersistentData( "myData" );
fl.trace("myData = "+ elt.getPersistentData("myData"));
}
```

## element.getTransformationPoint()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.getTransformationPoint()
```

### Parameters

Geen.

### Geretourneerde waarde

Een punt (bijvoorbeeld {x:10, y:20}, waarbij x en y drijvende-kommawaarden zijn) dat de positie van het transformatiepunt aangeeft (ook wel *oorsprong* of *nulpunt* genoemd) in het coördinatensysteem van het element.

### Beschrijving

Methode; hiermee wordt de waarde van het transformatiepunt van het opgegeven element opgehaald.

Transformatiepunten zijn relatief ten opzichte van verschillende locaties, afhankelijk van het type item dat is geselecteerd. Zie [element.setTransformationPoint\(\)](#) voor meer informatie.

### Voorbeeld

In het volgende voorbeeld wordt het transformatiepunt voor het derde element in het negende frame op de eerste laag in het document opgehaald. De eigenschap `transPoint.x` geeft de *x*-coördinaat van het transformatiepunt. De eigenschap `transPoint.y` bevat de *y*-coördinaat van het transformatiepunt.

```
var transPoint =
fl.getDocumentDOM().getTimeline().layers[0].frames[8].elements[2].getTransformationPoint();
```

### Zie ook

[document.getTransformationPoint\(\)](#), [element.setTransformationPoint\(\)](#), [element.transformX](#), [element.transformY](#)

## element.hasPersistentData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.hasPersistentData(name)
```

### Parameters

**name** Een tekenreeks waarmee de naam wordt opgegeven van het gegevensitem dat moet worden getest.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de opgegeven gegevens aan het object zijn gekoppeld, anders `false`.

### Beschrijving

Methode; hiermee wordt bepaald of de opgegeven gegevens aan het opgegeven element zijn gekoppeld. Alleen symbolen en bitmaps ondersteunen permanente gegevens.

### Voorbeeld

Zie `element.getPersistentData()`.

## element.height

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.height
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de hoogte van het element in pixels aangeeft.

Gebruik deze eigenschap niet om het formaat van een tekstveld te wijzigen. Selecteer in plaats daarvan het tekstveld en gebruik `document.setTextRectangle()`. Als u deze eigenschap in een tekstveld gebruikt, wordt de grootte van de tekst gewijzigd.

### Voorbeeld

In het volgende voorbeeld wordt de hoogte van het opgegeven element geschaald naar 100:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].height = 100;
```

## element.layer

### Beschikbaarheid

Flash 8.



### Gebruik

`element.layer`

### Beschrijving

Alleen-lezen eigenschap; staat voor het [Object Layer](#) waarop het element zich bevindt.

### Voorbeeld

In het volgende voorbeeld wordt het object Layer dat het element bevat, in de variabele `theLayer` opgeslagen:

```
var theLayer = element.layer;
```

## element.left

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`element.left`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die staat voor de linkerzijde van het element. De waarde van `element.left` is relatief ten opzichte van de linkerbovenhoek van het werkgebied voor elementen in een scène, en is relatief ten opzichte van het registratiepunt van het symbool (ook wel *oorsprong* of *nulpunt* genoemd) als het element binnen een symbool is opgeslagen. Gebruik [document.setSelectionBounds\(\)](#) of [document.moveSelectionBy\(\)](#) om deze eigenschap in te stellen.

### Voorbeeld

In het volgende voorbeeld wordt getoond hoe de waarde van deze eigenschap verandert wanneer een element wordt verplaatst:

```
// Select an element on the Stage and then run this script.  
var sel = fl.getDocumentDOM().selection[0];  
fl.trace("Left (before) = " + sel.left);  
fl.getDocumentDOM().moveSelectionBy({x:100, y:0});  
fl.trace("Left (after) = " + sel.left);
```

Zie het voorbeeld voor [element.elementType](#).

## element.locked

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`element.locked`

### Beschrijving

Eigenschap; een Booleaanse waarde: `true` als het element vergrendeld is, anders `false`. Als de waarde van `element.elementType"shape"` is, wordt deze eigenschap genegeerd.

### Voorbeeld

In het volgende voorbeeld wordt het eerste element in het eerste frame, bovenste laag vergrendeld:

```
// Similar to Modify > Arrange > Lock:  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].locked = true;
```

## element.matrix

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.matrix
```

### Beschrijving

Eigenschap; een object Matrix. Een matrix heeft de eigenschappen `a`, `b`, `c`, `d`, `tx` en `ty`. De eigenschappen `a`, `b`, `c` en `d` zijn drijvende-kommapwaarden; de eigenschappen `tx` en `ty` zijn coördinaten. Zie [Object Matrix](#).

### Voorbeeld

In het volgende voorbeeld wordt het opgegeven element 10 pixels verplaatst in `x` en 20 pixels in `y`:

```
var mat = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].matrix;  
mat.tx += 10;  
mat.ty += 20;  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].matrix = mat;
```

## element.name

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.name
```

### Beschrijving

Eigenschap; een tekenreeks die de naam van het element aanduidt. Wordt meestal de naam van de instantie genoemd. Als de waarde van `element.elementType"shape"` is, wordt deze eigenschap genegeerd. Zie [element.elementType](#).

### Voorbeeld

In het volgende voorbeeld wordt de naam van de instantie van het eerste element in Frame 1, bovenste laag ingesteld op `"clip_mc"`:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].name = "clip_mc";
```

Zie het voorbeeld voor `element.elementType`.

## element.removePersistentData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.removePersistentData(name)
```

### Parameters

**name** Een tekenreeks waarmee de naam wordt opgegeven van de gegevens die moeten worden verwijderd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden eventuele permanente gegevens met de opgegeven naam verwijderd die aan het object zijn gekoppeld. Alleen symbolen en bitmaps ondersteunen permanente gegevens.

### Voorbeeld

Zie `element.getPersistentData()`.

## element.rotation

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.rotation
```

### Beschrijving

Eigenschap; een geheel getal of drijvende-kommawaarde tussen -180 en 180 waarmee de draaiing (met de klok mee) van het object wordt aangegeven, in graden.

### Voorbeeld

In het volgende voorbeeld wordt de draaiing van het geselecteerde element op 45 graden ingesteld:

```
var element = fl.getDocumentDOM().selection[0];
fl.trace("Element rotation = " + element.rotation);
element.rotation = 45;
fl.trace("After setting rotation to 45: rotation = " + element.rotation);
```

## element.scaleX

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.scaleX
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de  $x$ -schaalwaarde van symbolen, tekenobjecten en primitieve rechthoeken en ovaal aanduidt. Een waarde 1 staat voor een schaling van 100 procent.

### Voorbeeld

In het volgende voorbeeld wordt de  $x$ -schaalwaarde van de huidige selectie ingesteld op 2 (verdubbelt de waarde):

```
var element = fl.getDocumentDOM().selection[0];  
element.scaleX = 2;
```

### Zie ook

[element.scaleY](#)

## element.scaleY

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.scaleY
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de  $y$ -schaalwaarde van symbolen, tekenobjecten en primitieve rechthoeken en ovaal aanduidt. Een waarde 1 staat voor een schaling van 100 procent.

### Voorbeeld

In het volgende voorbeeld wordt de  $y$ -schaalwaarde van de huidige selectie ingesteld op 2 (verdubbelt de waarde):

```
var element = fl.getDocumentDOM().selection[0];  
element.scaleY = 2;
```

### Zie ook

[element.scaleX](#)

## element.selected

### Beschikbaarheid

Flash 8.

### Gebruik

```
element.selected
```

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of het element moet worden geselecteerd (*true*) of niet (*false*).

### Voorbeeld

In het volgende voorbeeld wordt het element geselecteerd:

```
element.selected = true;
```

## element.setPersistentData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.setPersistentData(name, type, value)
```

### Parameters

**name** Een tekenreeks die de naam opgeeft die aan de gegevens moet worden gekoppeld. Deze naam wordt gebruikt om de gegevens op te halen.

**type** Een tekenreeks die het type gegevens aangeeft. De toegestane waarden zijn "integer", "integerArray", "double", "doubleArray", "string" en "byteArray".

**value** Hiermee wordt de waarde opgegeven die aan het object moet worden gekoppeld. Het gegevenstype van *value* hangt af van de waarde van de parameter *type*. De opgegeven waarde moet zijn geschikt voor het gegevenstype dat wordt aangegeven door de parameter *type*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden gegevens met een element opgeslagen. De gegevens zijn beschikbaar wanneer het FLA-bestand met het element opnieuw wordt geopend. Alleen symbolen en bitmaps ondersteunen permanente gegevens.

### Voorbeeld

Zie `element.getPersistentData()`.

## element.setTransformationPoint()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.setTransformationPoint(transformationPoint)
```

### Parameters

**transformationPoint** Een punt (bijvoorbeeld {x:10, y:20}, waarbij x en y drijvende-kommawaarden zijn) dat waarden aangeeft voor het transformatiepunt van een element of groep.

- Vormen: *transformationPoint* wordt ingesteld ten opzichte van het document (0,0 is de linkerbovenhoek van het werkgebied).
- Symbolen: *transformationPoint* wordt ingesteld ten opzichte van het registratiepunt van het symbool (0,0 bevindt zich op het registratiepunt).
- Tekst: *transformationPoint* wordt ten opzichte van het tekstveld ingesteld (0,0 is de linkerbovenhoek van het tekstveld).
- Bitmaps/video's: *transformationPoint* wordt ingesteld ten opzichte van de bitmap/video (0,0 is de linkerbovenhoek van de bitmap of video).
- Tekenobjecten, primitieve objecten en groepen: *transformationPoint* wordt ingesteld ten opzichte van het middelpunt van het element of de groep (0,0 is het middelpunt van het element of de groep).

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de positie ingesteld van het transformatiepunt van het opgegeven element.

Deze methode is vrijwel gelijk aan `document.setTransformationPoint()`. De methode kent de volgende verschillen:

- Het transformatiepunt voor tekenobjecten, primitieve objecten en groepen wordt ingesteld ten opzichte van het middelpunt van het element of de groep, niet ten opzichte van het werkgebied.
- U kunt transformatiepunten voor elementen instellen zonder deze eerst te selecteren.

Met deze methode verplaatst u het transformatiepunt, maar niet het element. Met de eigenschappen `element.transformX` en `element.transformY` daarentegen wordt het element verplaatst.

### Voorbeeld

In het volgende voorbeeld wordt het transformatiepunt van het derde element op het werkgebied ingesteld op 100, 200:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[2].setTransformationPoint({x:100, y:200});
```

### Zie ook

`document.setTransformationPoint().element.getTransformationPoint()`, `element.transformX`, `element.transformY`

## element.skewX

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.skewX
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde tussen -180 en 180 die de  $x$ -waarde voor scheeftrekken van symbolen, tekenobjecten en primitieve rechthoeken en ovaal aanduidt.

### Voorbeeld

In het volgende voorbeeld wordt de  $x$ -waarde voor scheeftrekken van de huidige selectie ingesteld op 10:

```
var element = fl.getDocumentDOM().selection[0];  
element.skewX = 10;
```

### Zie ook

[document.setTransformationPoint\(\)](#), [element.skewY](#)

## element.skewY

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.skewY
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde tussen -180 en 180 die de  $y$ -waarde voor scheeftrekken van symbolen, tekenobjecten en primitieve rechthoeken en ovaal aanduidt.

### Voorbeeld

In het volgende voorbeeld wordt de  $y$ -waarde voor scheeftrekken van de huidige selectie ingesteld op 10:

```
var element = fl.getDocumentDOM().selection[0];  
element.skewY = 10;
```

### Zie ook

[document.setTransformationPoint\(\)](#), [element.skewX](#)

## element.top

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`element.top`

### Beschrijving

Alleen-lezen eigenschap; bovenzijde van het element. De waarde van `element.top` is relatief ten opzichte van de linkerbovenhoek van het werkgebied voor elementen in een scène, en is relatief ten opzichte van het registratiepunt van het symbool als het element binnen een symbool is opgeslagen. Gebruik `document.setSelectionBounds()` of `document.moveSelectionBy()` om deze eigenschap in te stellen.

### Voorbeeld

In het volgende voorbeeld wordt getoond hoe de waarde van deze eigenschap verandert wanneer een element wordt verplaatst:

```
// Select an element on the Stage and then run this script.  
var sel = fl.getDocumentDOM().selection[0];  
fl.trace("Top (before) = " + sel.top);  
fl.getDocumentDOM().moveSelectionBy({x:0, y:100});  
fl.trace("Top (after) = " + sel.top);
```

Zie het voorbeeld voor `element.elementType`.

## element.transformX

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`element.transformX`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de  $x$ -waarde van het transformatiepunt van het geselecteerde element aanduidt, binnen het coördinatensysteem van het bovenliggende element van het element. Als u voor deze eigenschap een nieuwe waarde instelt, wordt het element verplaatst. Met de methode `element.setTransformationPoint()` daarentegen wordt het transformatiepunt verplaatst, maar niet het element.

### Voorbeeld

### Zie ook

`element.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformY`



## element.transformY

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.transformY
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de *y*-waarde van het transformatiepunt van het geselecteerde element aanduidt, binnen het coördinatensysteem van het bovenliggende element van het element. Als u voor deze eigenschap een nieuwe waarde instelt, wordt het element verplaatst. Met de methode `element.setTransformationPoint()` daarentegen wordt het transformatiepunt verplaatst, maar niet het element.

### Zie ook

`element.getTransformationPoint()`, `element.setTransformationPoint()`, `element.transformX`

## element.width

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
element.width
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de breedte van het element in pixels aangeeft.

Gebruik deze eigenschap niet om het formaat van een tekstveld te wijzigen. Selecteer in plaats daarvan het tekstveld en gebruik `document.setTextRectangle()`. Als u deze eigenschap in een tekstveld gebruikt, wordt de grootte van de tekst gewijzigd.

### Voorbeeld

In het volgende voorbeeld wordt de breedte van het opgegeven element geschaald naar 100:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].width= 100;
```

## element.x

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
element.x
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de *x*-waarde van het registratiepunt van het geselecteerde element aanduidt.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van het registratiepunt van het opgegeven element ingesteld op 100, 200:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].x= 100;  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].y= 200;
```

### Zie ook

[element.y](#)

## element.y

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`element.y`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de *y*-waarde van het registratiepunt van het geselecteerde element aanduidt.

### Voorbeeld

Zie [element.x](#)

# Hoofdstuk 15: Object Fill

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Fill bevat alle eigenschappen van de vulkleurinstelling van het deelvenster Gereedschappen of van een geselecteerde vorm. Gebruik `document.getCustomFill()` om een object Fill op te halen.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Fill:

Eigenschap	Beschrijving
<code>fill.bitmapIsClipped</code>	Een Booleaanse waarde die opgeeft of het bitmapbestand voor een vorm die groter is dan de bitmap, wordt bijgesneden of herhaald.
<code>fill.bitmapPath</code>	Een tekenreeks die het pad en de naam van de bitmapvulling in de bibliotheek opgeeft.
<code>fill.color</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat de vulkleur vertegenwoordigt.
<code>fill.colorArray</code>	Een array met kleuren in een verloop.
<code>fill.focalPoint</code>	Een geheel getal dat de horizontale verschuiving van het brandpunt van het verloop opgeeft vanaf het transformatiepunt.
<code>fill.linearRGB</code>	Een Booleaanse waarde die opgeeft of de vulling moet worden gerenderd als lineair of radiaal RGB-verloop.
<code>fill.matrix</code>	Een <a href="#">Object Matrix</a> dat de plaatsing, oriëntatie en schaal voor vullingen met verloop vastlegt.
<code>fill.overflow</code>	Een tekenreeks die het gedrag van de overflow van een verloop opgeeft.
<code>fill.posArray</code>	Een array met gehele getallen, elk in het bereik van 0 tot en met 255, die de positie van de corresponderende kleur aangeven.
<code>fill.style</code>	Een tekenreeks die de vulstijl opgeeft.

## fill.bitmapsClipped

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fill.bitmapIsClipped
```

### Beschrijving

Eigenschap: een Booleaanse waarde die opgeeft of het bitmapbestand voor een vorm die groter is dan de bitmap, wordt bijgesneden (`true`) of herhaald (`false`). Deze eigenschap is alleen beschikbaar als de waarde van de eigenschap `fill.style` "bitmap" is. Als de vorm kleiner is dan de bitmap, is deze waarde `false`.

### Voorbeeld

In het volgende voorbeeld, wordt indien van toepassing, in het deelvenster Uitvoer weergegeven of de bitmapvulling is bijgesneden:

```
var fill = fl.getDocumentDOM().getCustomFill();  
if (fill.style == "bitmap")  
    fl.trace("Fill image is clipped: " + fill.bitmapIsClipped);
```

### Zie ook

[fill.bitmapPath](#)

## fill.bitmapPath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`fill.bitmapPath`

### Beschrijving

Eigenschap: een tekenreeks die het pad en de naam van de bitmapvulling in de bibliotheek opgeeft. Deze eigenschap is alleen beschikbaar als de waarde van de eigenschap `fill.style` "bitmap" is.

### Voorbeeld

In het volgende voorbeeld wordt de vulstijl van het opgegeven item ingesteld op een bitmapafbeelding in de bibliotheek:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.style = "bitmap";  
fill.bitmapPath = "myBitmap.jpg";  
fl.getDocumentDOM().setCustomFill(fill);
```

### Zie ook

[fill.bitmapIsClipped](#)

## fill.color

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`fill.color`

### Beschrijving

Eigenschap; de kleur van de vulling in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"

**Object Fill**

- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

**Voorbeeld**

In het volgende voorbeeld wordt de vulkleur van de huidige selectie ingesteld:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.color = "#FFFFFF";
fl.getDocumentDOM().setCustomFill( fill );
```

## fill.colorArray

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
fill.colorArray
```

**Beschrijving**

Eigenschap; een array met kleuren in het verloop, uitgedrukt als gehele getallen. Deze eigenschap is alleen beschikbaar als de waarde van de eigenschap `fill.style` "radialGradient" of "linearGradient" is. Zie [fill.style](#)

**Voorbeeld**

In het volgende voorbeeld wordt, indien van toepassing, in het deelvenster Uitvoer de kleurenmatrix van de huidige selectie weergegeven:

```
var fill = fl.getDocumentDOM().getCustomFill();
if(fill.style == "linearGradient" || fill.style == "radialGradient")
    alert(fill.colorArray);
```

In het volgende voorbeeld wordt de vulling ingesteld op het opgegeven lineaire verloop:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "linearGradient";
fill.colorArray = ["#00ff00", "#ff00ff"];
fill.posArray = [0, 255];
fl.getDocumentDOM().setCustomFill(fill);
```

## fill.focalPoint

**Beschikbaarheid**

Flash 8.

**Gebruik**

```
fill.focalPoint
```

### Beschrijving

Eigenschap; een geheel getal dat de horizontale verschuiving van het brandpunt van het verloop opgeeft vanaf het transformatiepunt. De waarde 10 zou bijvoorbeeld het brandpunt plaatsen op 10/255 van de afstand van het transformatiepunt naar de rand van het verloop. De waarde -255 zou het brandpunt plaatsen op de linkergrens van het verloop. De standaardwaarde is 0.

Deze eigenschap is alleen beschikbaar als de waarde van de eigenschap `fill.style` "radialGradient" is.

### Voorbeeld

In het volgende voorbeeld wordt het brandpunt van een radiaal verloop voor de huidige selectie ingesteld op 100 pixels rechts van het midden van de vorm:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "radialGradient";
fill.colorArray = ["#00ff00", "#ff00ff"];
fill.posArray = [0, 255];
fill.focalPoint = 10100;
fl.getDocumentDOM().setCustomFill(fill);
```

## fill.linearRGB

### Beschikbaarheid

Flash 8.

### Gebruik

```
fill.linearRGB
```

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of de vulling moet worden gerenderd als lineair of radiaal RGB-verloop. Stel deze eigenschap in op `true` om een lineaire interpolatie van een verloop op te geven; stel deze in op `false` om een radiale interpolatie van een verloop op te geven. De standaardwaarde is `false`.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat het verloop van de huidige selectie moet worden gerenderd met een lineaire RGB:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.linearRGB style = true"radialGradient";
fill.colorArray = ["#00ff00", "#ff00ff"];
fill.posArray = [0, 255];
fill.focalPoint = 100;
fill.linearRGB = true;
fl.getDocumentDOM().setCustomFill(fill);
```

## fill.matrix

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`fill.matrix`

### Beschrijving

Eigenschap; een [Object Matrix](#) dat de plaatsing, oriëntatie en schaal voor vullingen met verloop vastlegt.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `fill.matrix` gebruikt om een vulling met verloop op te geven voor de huidige selectie:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = 'radialGradient';
fill.colorArray = ['#00ff00', '#ff00ff'];
fill.posArray = [0, 255];
fill.focalPoint = 100;
fill.linearRGB = false;
fill.overflow = 'repeat';
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 0.0167083740234375;
mat.b = -0.0096435546875;
mat.c = 0.0312957763671875;
mat.d = 0.05419921875;
mat.tx = 288.65;
mat.ty = 193.05;
for (i in mat) {
    fl.trace(i+' : '+mat[i]);
}
fl.getDocumentDOM().setCustomFill(fill);
```

## fill.overflow

### Beschikbaarheid

Flash 8.

### Gebruik

`fill.overflow`

### Beschrijving

Eigenschap; een tekenreeks die het gedrag van de overloop van een verloop opgeeft. Acceptabele waarden zijn "extend", "repeat" en "reflect"; de tekenreeksen zijn niet hoofdlettergevoelig. De standaardwaarde is "extend".

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat het gedrag van de overloop voor de huidige selectie "extend" moet zijn:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.overflow = "extend";
fl.getDocumentDOM().setCustomFill(fill);
```

## fill.posArray

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`fill.posArray`

### Beschrijving

Eigenschap; een array van gehele getallen, elk in het bereik van 0 tot en met 255, de die positie van de corresponderende kleur aangeven. Deze eigenschap is alleen beschikbaar als de waarde van de eigenschap

`fill.style` "radialGradient" of "linearGradient" is.

### Voorbeeld

In het volgende voorbeeld worden de kleuren opgegeven die moeten worden gebruikt in een lineair verloop voor de huidige selectie:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "linearGradient";
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];
fill.posArray= [0,100, 200];
fl.getDocumentDOM().setCustomFill( fill );
```

## fill.style

### Beschikbaarheid

Flash MX 2004. Waarde "bitmap" toegevoegd in Flash CS4 Professional.

### Gebruik

`fill.style`

### Beschrijving

Eigenschap; een tekenreeks die de vulstijl opgeeft. Acceptabele waarden zijn "bitmap", "solid", "linearGradient", "radialGradient" en "noFill".

Als deze waarde "linearGradient" of "radialGradient" is, zijn de eigenschappen `fill.colorArray` en `fill.posArray` ook beschikbaar. Als deze waarde "bitmap" is, zijn de eigenschappen `fill.bitmapIsClipped` en `fill.bitmapPath` ook beschikbaar.

### Voorbeeld

In het volgende voorbeeld worden de kleuren opgegeven die moeten worden gebruikt in een lineair verloop voor de huidige selectie:

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style= "linearGradient";
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];
fill.posArray= [0,100, 200];
fl.getDocumentDOM().setCustomFill( fill );
```



# Hoofdstuk 16: Object Filter

## Beschikbaarheid

Flash 8.

## Beschrijving

Dit object bevat alle eigenschappen voor alle filters. De eigenschap `filter.name` geeft het type filter aan en bepaalt welke eigenschappen op elke filter van toepassing zijn. Zie `filter.name`.

Gebruik `document.getFilters()` om de lijst met filters voor één of meerdere objecten te retourneren. Gebruik `document.setFilters()` om filters op één of meerdere objecten toe te passen. Zie `document.getFilters()` en `document.setFilters()`.

## Overzicht van eigenschappen

U kunt de volgende methoden gebruiken met het object Filter:

Eigenschap	Beschrijving
<code>filter.angle</code>	Een drijvende-kommawaarde die de hoek van de schaduw- of markeerleur in graden aangeeft.
<code>filter.blurX</code>	Een drijvende-kommawaarde die de vervaging in de x-richting in pixels aangeeft.
<code>filter.blurY</code>	Een drijvende-kommawaarde die de vervaging in de y-richting in pixels aangeeft.
<code>filter.brightness</code>	Een drijvende-kommawaarde die de helderheid van de filter aangeeft.
<code>filter.color</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat de filter kleur vertegenwoordigt.
<code>filter.contrast</code>	Een drijvende-kommawaarde die de contrastwaarde van de filter aangeeft.
<code>filter.distance</code>	Een drijvende-kommawaarde die de afstand tussen het effect van de filter en een object in pixels aangeeft.
<code>filter.enabled</code>	Een Booleaanse waarde die aangeeft of de opgegeven filter is ingeschakeld.
<code>filter.hideObject</code>	Een Booleaanse waarde die aangeeft of de bronafbeelding is verborgen.
<code>filter.highlightColor</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat de markeerleur vertegenwoordigt.
<code>filter.hue</code>	Een drijvende-kommawaarde die de kleurtoon van de filter aangeeft.
<code>filter.inner</code>	Een Booleaanse waarde die aangeeft of de schaduw een binnenschaduw is.
<code>filter.knockout</code>	Een Booleaanse waarde die aangeeft of de filter uitneembaar is.
<code>filter.name</code>	Alleen-lezen; een tekenreeks die het type filter aangeeft.
<code>filter.quality</code>	Een tekenreeks die de kwaliteit van de vervaging aangeeft.
<code>filter.saturation</code>	Een drijvende-kommawaarde die de verzadigingswaarde van de filter aangeeft.
<code>filter.shadowColor</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat de schaduwkleur vertegenwoordigt.
<code>filter.strength</code>	Een geheel getal dat het sterktepercentage van de filter aangeeft.
<code>filter.type</code>	Een tekenreeks die het type schuine kant of gloed aangeeft.

## filter.angle

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.angle`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de hoek van de schaduw- of markeerkleur in graden aangeeft. Acceptabele waarden liggen tussen 0 en 360. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "dropShadowFilter", "gradientBevelFilter" of "gradientGlowFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de hoek voor de filters Schuine kant op het/de geselecteerde object(en) ingesteld op 120:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++) {
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].angle = 120;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

`document.setFilterProperty()`

## filter.blurX

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.blurX`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de vervaging in de *x*-richting in pixels aangeeft. Acceptabele waarden liggen tussen 0 en 255. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" of "gradientGlowFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld worden de waarden `blurX` en `blurY` voor de filters Vaag op het/de geselecteerde object(en) ingesteld op respectievelijk 30 en 20:

**Object Filter**

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'blurFilter'){
        myFilters[i].blurX = 30;
        myFilters[i].blurY = 20;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

**Zie ook**

[document.setFilterProperty\(\)](#), [filter.blurY](#)

## filter.blurY

**Beschikbaarheid**

Flash 8.

**Gebruik**

`filter.blurY`

**Beschrijving**

Eigenschap; een drijvende-kommawaarde die de vervaging in de *y*-richting in pixels aangeeft. Acceptabele waarden liggen tussen 0 en 255. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientBevelFilter" of "gradientGlowFilter" voor de eigenschap `filter.name`.

**Voorbeeld**

Zie [filter.blurX](#).

**Zie ook**

[document.setFilterProperty\(\)](#), [filter.blurX](#)

## filter.brightness

**Beschikbaarheid**

Flash 8.

**Gebruik**

`filter.brightness`

**Beschrijving**

Eigenschap; een drijvende-kommawaarde die de helderheid van de filter aangeeft. Acceptabele waarden liggen tussen -100 en 100. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "adjustColorFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de helderheid voor de filters Kleur aanpassen op het/de geselecteerde object(en) ingesteld op 30.5:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].brightness = 30.5;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

## filter.color

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.color`

### Beschrijving

Eigenschap; de kleur van de filter, in één van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "dropShadowFilter" of "glowFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de kleur voor de filters Slagschaduw op het/de geselecteerde object(en) ingesteld op "#ff00003e":

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].color = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

`document.setFilterProperty()`

## filter.contrast

### Beschikbaarheid

Flash 8.

### Gebruik

```
filter.contrast
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de contrastwaarde van de filter aangeeft. Acceptabele waarden liggen tussen -100 en 100. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "adjustColorFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de contrastwaarde voor de filters Kleur aanpassen op het/de geselecteerde object(en) ingesteld op -15.5:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].contrast = -15.5;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

## filter.distance

### Beschikbaarheid

Flash 8.

### Gebruik

```
filter.distance
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de afstand tussen het effect van de filter en een object in pixels aangeeft. Acceptabele waarden liggen tussen -255 en 255. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "dropShadowFilter", "gradientBevelFilter" of "gradientGlowFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de afstand voor de filters Slagschaduw op het/de geselecteerde object(en) ingesteld op 10 pixels:

**Object Filter**

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].distance = 10;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

**Zie ook**

[document.setFilterProperty\(\)](#)

## filter.enabled

**Beschikbaarheid**

Flash CS3 Professional.

**Gebruik**

`filter.enabled`

**Beschrijving**

Eigenschap; een Booleaanse waarde die aangeeft of de opgegeven filter is in- (`true`) of uitgeschakeld (`false`).

**Voorbeeld**

In het volgende voorbeeld worden de filters Kleur op het/de geselecteerde object(en) uitgeschakeld:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].enabled = false;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

## filter.hideObject

**Beschikbaarheid**

Flash 8.

**Gebruik**

`filter.hideObject`

**Beschrijving**

Eigenschap; een Booleaanse waarde die aangeeft of de bronafbeelding wordt verborgen (`true`) of weergegeven (`false`). Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "dropShadowFilter" voor de eigenschap [filter.name](#).

### Voorbeeld

In het volgende voorbeeld wordt de waarde `hideObject` voor de filters Slagschaduw op het/de geselecteerde object(en) ingesteld op `true`:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].hideObject = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

## filter.highlightColor

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.highlightColor`

### Beschrijving

Eigenschap; de kleur van de markering, in één van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de markeerkleur voor de filters Schuine kant op het/de geselecteerde object(en) ingesteld op "#ff00003e":

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].highlightColor = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

## filter.hue

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.hue`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de kleurtoon van de filter aangeeft. Acceptabele waarden liggen tussen -180 en 180. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "adjustColorFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de kleurtoon voor de filters Kleur aanpassen op het/de geselecteerde object(en) ingesteld op 120:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].hue = 120;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

## filter.inner

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.inner`

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de schaduw een binnenschaduw is (`true`) of niet (`false`). Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "dropShadowFilter" of "glowFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `inner` voor de Gloedfilters op het/de geselecteerde object(en) ingesteld op `true`:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].inner = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

[document.setFilterProperty\(\)](#)



## filter.knockout

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.knockout`

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de filter uitneembaar is (`true`) of niet (`false`). Deze eigenschap is gedefinieerd voor objecten `Filter` met de waarden `"bevelFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` of `"gradientGlowFilter"` voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `knockout` voor de Gloedfilters op het/de geselecteerde object(en) ingesteld op `true`:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].knockout = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

[document.setFilterProperty\(\)](#)

## filter.name

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.name`

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het type filter aangeeft. De waarde van deze eigenschap bepaalt welke andere eigenschappen van het object `Filter` beschikbaar zijn. Mogelijke waarden zijn: `"adjustColorFilter"`, `"bevelFilter"`, `"blurFilter"`, `"dropShadowFilter"`, `"glowFilter"`, `"gradientBevelFilter"` of `"gradientGlowFilter"`.

### Voorbeeld

In het volgende voorbeeld ziet u de namen van de filters en de indexposities in het deelvenster Uitvoer:

**Object Filter**

```
var myFilters = fl.getDocumentDOM().getFilters();
var traceStr = "";
for(i=0; i < myFilters.length; i++){
    traceStr = traceStr + " At index " + i + ": " + myFilters[i].name;
}
fl.trace(traceStr);
```

**Zie ook**

[document.getFilters\(\)](#), [document.setFilterProperty\(\)](#)

## filter.quality

**Beschikbaarheid**

Flash 8.

**Gebruik**

`filter.quality`

**Beschrijving**

Eigenschap; een tekenreeks die de kwaliteit van de vervaging aangeeft. Acceptabele waarden zijn "low", "medium" en "high" ("high" is vergelijkbaar met Gaussiaans vervagen). Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "blurFilter", "dropShadowFilter", "glowFilter", "gradientGlowFilter" of "gradientBevelFilter" voor de eigenschap [filter.name](#).

**Voorbeeld**

In het volgende voorbeeld wordt de kwaliteit van de vervaging voor de Gloedfilters op het/de geselecteerde object(en) ingesteld op "medium":

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].quality = 'medium';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

**Zie ook**

[document.setFilterProperty\(\)](#)

## filter.saturation

**Beschikbaarheid**

Flash 8.

**Gebruik**

`filter.saturation`

### Beschrijving

Eigenschap; een drijvende-komawaaarde die de verzadigingswaarde van de filter aangeeft. Acceptabele waarden liggen tussen -100 en 100. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "adjustColorFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de verzadigingswaarde voor de filters Kleur aanpassen op het/de geselecteerde object(en) ingesteld op -100 (grijswaarden):

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].saturation = 0-100;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

`document.setFilterProperty()`

## filter.shadowColor

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.shadowColor`

### Beschrijving

Eigenschap; de kleur van de schaduw in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter" voor de eigenschap `filter.name`.

### Voorbeeld

In het volgende voorbeeld wordt de schaduwkleur voor de filters Schuine kant op het/de geselecteerde object(en) ingesteld op "#ff00003e":

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].shadowColor = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

[document.setFilterProperty\(\)](#)

## filter.strength

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.strength`

### Beschrijving

Eigenschap; een geheel getal dat het sterktepercentage van de filter aangeeft. Acceptabele waarden liggen tussen 0 en 25,500. Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "dropShadowFilter", "glowFilter", "gradientGlowFilter" of "gradientBevelFilter" voor de eigenschap [filter.name](#).

### Voorbeeld

In het volgende voorbeeld wordt de sterkte voor de Gloedfilters op het/de geselecteerde object(en) ingesteld op 50:

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].strength = 50;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

### Zie ook

[document.setFilterProperty\(\)](#)

## filter.type

### Beschikbaarheid

Flash 8.

### Gebruik

`filter.type`

### Beschrijving

Eigenschap; een tekenreeks die het type schuine kant of gloed aangeeft. Acceptabele waarden zijn "inner", "outer" en "full". Deze eigenschap is gedefinieerd voor objecten Filter met de waarde "bevelFilter", "gradientGlowFilter" of "gradientBevelFilter" voor de eigenschap [filter.name](#).

### Voorbeeld

In het volgende voorbeeld wordt het type voor alle filters Schuine kant op het/de geselecteerde object(en) ingesteld op "full":

**Object Filter**

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].type = 'full';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

**Zie ook**

[document.setFilterProperty\(\)](#)

# Hoofdstuk 17: Object flash (fl)

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object flash vertegenwoordigt de toepassing Flash. U kunt `flash` of `fl` gebruiken om naar dit object te verwijzen. Overal in deze documentatie wordt `fl` gebruikt in voorbeeldcode.

## Overzicht van methoden

Voor het object flash kunnen de volgende methoden worden gebruikt:

Methoden	Beschrijving
<code>fl.addEventListener()</code>	Hiermee wordt een functie geregistreerd die moet worden aangeroepen wanneer een bepaalde gebeurtenis is ontvangen.
<code>fl.browseForFileURL()</code>	Hiermee wordt het systeemdialogvenster Bestand openen of Bestand opslaan geopend waarin de gebruiker een bestand kan opgeven dat moet worden geopend of opgeslagen.
<code>fl.browseForFolderURL()</code>	Hiermee wordt het dialogvenster Bladeren naar map geopend waarin de gebruiker een map kan selecteren.
<code>fl.clearPublishCache()</code> op pagina 230	Hiermee wordt de publicatiecache gewist.
<code>fl.clipCopyString()</code>	Hiermee wordt de opgegeven tekenreeks naar het klembord gekopieerd.
<code>fl.closeAll()</code>	Hiermee worden alle geopende documenten gesloten en wordt het dialogvenster Opslaan als geopend voor alle documenten die nog niet zijn opgeslagen.
<code>fl.closeAllPlayerDocuments()</code>	Hiermee worden alle SWF-bestanden gesloten die zijn geopend met Besturing > Film testen.
<code>fl.closeDocument()</code>	Hiermee wordt het opgegeven document gesloten.
<code>fl.createDocument()</code>	Hiermee wordt een nieuw document geopend en geselecteerd.
<code>fl.exportPublishProfileString()</code> op pagina 238	Uniform Resource Identifier (URI) waaruit publicatie-instellingen moeten worden geëxporteerd.
<code>fl.fileExists()</code>	Hiermee wordt gecontroleerd of een bestand al op schijf aanwezig is.
<code>fl.findDocumentDOM()</code>	Hiermee kunt u een specifiek bestand als doel opgeven met zijn unieke id.
<code>fl.findDocumentIndex()</code>	Hiermee wordt een array met gehele getallen geretourneerd die staan voor de positie van een document in de array <code>fl.documents</code> .
<code>fl.findObjectInDocByName()</code>	Hiermee worden elementen gezocht met instantienamen die met opgegeven tekst overeenkomen.
<code>fl.findObjectInDocByType()</code>	Hiermee worden elementen van een specifiek elementtype gezocht in een document.
<code>fl.getAppMemoryInfo()</code>	Hiermee wordt een geheel getal geretourneerd dat staat voor het aantal bytes dat in een bepaald gebied van het Flash.exe-geheugen wordt gebruikt.

Methode	Beschrijving
<code>fl.getDocumentDOM()</code>	Hiermee wordt de DOM ( <b>Object Document</b> ) van het actieve document opgehaald.
<code>fl.getSwfPanel()</code>	Hiermee wordt het SWFPanel-object geretourneerd op basis van de gelokaliseerde naam of de SWF-bestandsnaam van het deelvenster.
<code>fl.isFontInstalled()</code>	Hiermee wordt vastgesteld of een opgegeven lettertype is geïnstalleerd.
<code>fl.mapPlayerURL()</code>	Hiermee wordt een Unicode URL met escape-teken toegewezen aan een UTF-8 of MBCS URL.
<code>fl.openDocument()</code>	Hiermee wordt een Flash-document (FLA-bestand) geopend in een nieuw Flash-documentvenster om het te bewerken. Het krijgt de focus.
<code>fl.openScript()</code>	Hiermee wordt een script (JSFL, AS, ASC) of een ander bestand (XML, TXT) geopend in de teksteditor van Flash.
<code>fl.quit()</code>	Hiermee wordt Flash afgesloten en wordt de gebruiker gevraagd eventueel gewijzigde documenten op te slaan.
<code>fl.reloadTools()</code>	Hiermee wordt het deelvenster Gereedschappen opnieuw opgebouwd op basis van het bestand toolconfig.xml. Wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.
<code>fl.removeEventListener()</code>	Hiermee wordt de registratie van een functie ongedaan gemaakt die met <code>fl.addEventListener()</code> was geregistreerd.
<code>fl.resetAS3PackagePaths()</code>	Hiermee wordt de globale Classpath-instelling in het dialoogvenster ActionScript 3,0-instellingen teruggezet op de standaardwaarde.
<code>fl.resetPackagePaths()</code>	Hiermee wordt de globale Classpath-instelling in het dialoogvenster ActionScript 2.0-instellingen teruggezet op de standaardwaarde.
<code>fl.runScript()</code>	Hiermee wordt een JavaScript-bestand uitgevoerd.
<code>fl.saveAll()</code>	Hiermee worden alle geopende documenten opgeslagen en wordt het dialoogvenster Opslaan als geopend voor alle documenten die nog niet zijn opgeslagen.
<code>fl.saveDocument()</code>	Hiermee wordt het opgegeven document opgeslagen als FLA-document.
<code>fl.saveDocumentAs()</code>	Hiermee wordt het dialoogvenster Opslaan als weergegeven voor het opgegeven document.
<code>fl.selectElement()</code>	Hiermee wordt de selectie of bewerking van een element ingeschakeld.
<code>fl.selectTool()</code>	Hiermee wordt het opgegeven gereedschap geselecteerd in het deelvenster Gereedschappen.
<code>fl.setActiveWindow()</code>	Hiermee wordt het actieve venster ingesteld op het opgegeven document.
<code>fl.showIdleMessage()</code>	Hiermee kunt u de waarschuwing uitschakelen met betrekking tot een script dat te lang wordt uitgevoerd.
<code>fl.toggleBreakpoint()</code>	Hiermee wordt op de desbetreffende regel een onderbrekingspunt voor het desbetreffende AS-bestand in- en uitgeschakeld.
<code>fl.trace()</code>	Hiermee wordt een tekstreeks verzonden naar het deelvenster Uitvoer.

### Overzicht van eigenschappen

U kunt de volgende eigenschappen gebruiken met het object flash.

Eigenschap	Beschrijving
<code>fl.actionsPanel</code>	Alleen-lezen; een <a href="#">Object actionsPanel</a> .
<code>fl.as3PackagePaths</code>	Een tekenreeks die overeenkomt met de algemene instelling voor Classpath in het dialoogvenster ActionScript 3,0-instellingen.
<code>fl.compilerErrors</code>	Alleen-lezen; een <a href="#">Object compilerErrors</a> .
<code>fl.componentsPanel</code>	Alleen-lezen; een <a href="#">Object componentsPanel</a> dat staat voor het deelvenster Componenten.
<code>fl.configDirectory</code>	Alleen-lezen; een tekenreeks met het volledige pad voor de map Configuration van de lokale gebruiker als platformspecifiek pad.
<code>fl.configURI</code>	Alleen-lezen; een tekenreeks met het volledige pad voor de map Configuration van de lokale gebruiker als bestand:/// URI.
<code>fl.contactSensitiveSelection</code>	Een Booleaanse waarde die opgeeft of de selectiemodus Contactgevoelig is ingeschakeld.
<code>fl.createNewDocList</code>	Alleen-lezen; een array met tekenreeksen die staan voor de verschillende typen documenten die kunnen worden gemaakt.
<code>fl.createNewDocListType</code>	Alleen-lezen; een array met tekenreeksen die staan voor de bestandsextensies van de typen documenten die kunnen worden gemaakt.
<code>fl.createNewTemplateList</code>	Alleen-lezen; een array met tekenreeksen die staan voor de verschillende typen sjablonen die kunnen worden gemaakt.
<code>fl.documents</code>	Alleen-lezen; een array met objecten Document (zie <a href="#">Object Document</a> ) die staan voor de documenten (FLA-bestanden) die momenteel geopend zijn en kunnen worden bewerkt.
<code>fl.drawingLayer</code>	Alleen-lezen; het <a href="#">Object drawingLayer</a> dat een uitbreidbaar gereedschap moet gebruiken wanneer de gebruiker tijdens het slepen tijdelijk wil tekenen.
<code>fl.externalLibraryPath</code>	Een tekenreeks die een lijst met items bevat in het globale pad voor de externe bibliotheek van ActionScript 3.0. Hiermee wordt de locatie van SWC-bestanden aangegeven die worden gebruikt als gedeelde runtime bibliotheken.
<code>fl.flexSDKPath</code>	Een tekenreeks die het pad naar de Flex-map SDK opgeeft; deze map bevat de map bin, frameworks, lib en andere mappen.
<code>fl.installedPlayers</code>	Hiermee wordt een array van generieke objecten geretourneerd die overeenkomen met de lijst geïnstalleerde Flash Players in de Eigenschapcontrole van het document.
<code>fl.languageCode</code>	Hiermee wordt de code van vijf tekens geretourneerd die de locale aangeeft van de gebruikersinterface van de toepassing.
<code>fl.libraryPath</code>	Een tekenreeks die een lijst met items bevat in het globale pad voor de bibliotheek van ActionScript 3.0. Hiermee wordt de locatie aangegeven van SWC-bestanden of mappen die SWC-bestanden bevatten.
<code>fl.Math</code>	Alleen-lezen; het <a href="#">Object Math</a> biedt methoden voor matrix- en puntbewerkingen.
<code>fl.mruRecentFileList</code>	Alleen-lezen; een array met de volledige bestandsnamen in de MRU-lijst (Most Recently Used, laatst gebruikt) die in het Flash-ontwerpgereedschap wordt beheerd.
<code>fl.mruRecentFileListType</code>	Alleen-lezen; een array met de bestandstypen in de MRU-lijst die in het Flash-ontwerpgereedschap wordt beheerd.
<code>fl.packagePaths</code>	Een tekenreeks die overeenkomt met de algemene instelling voor Classpath in het dialoogvenster ActionScript 2.0-instellingen.



Eigenschap	Beschrijving
<code>fl.publishCacheDiskSizeMax</code>	Een geheel getal dat de voorkeur voor de limiet van de schijfcache instelt.
<code>fl.publishCacheEnabled</code>	Een booleaanse waarde die instelt of de publicatiecache ingeschakeld is.
<code>fl.publishCacheMemoryEntrySizeLimit</code>	Een geheel-getaleigenschap die de maximale limiet van de voorkeur voor de geheugencachevermelding instelt.
<code>fl.publishCacheMemorySizeMax</code>	Een geheel getal dat de voorkeur voor de limiet van de geheugencache instelt.
<code>fl.objectDrawingMode</code>	Een geheel getal dat staat voor de ingeschakelde objecttekenmodus.
<code>fl.outputPanel</code>	Alleen-lezen; verwijzing naar het <a href="#">Object outputPanel</a> .
<code>fl.presetPanel</code>	Alleen-lezen; een <a href="#">Object presetPanel</a> .
<code>fl.scriptURI</code>	Alleen-lezen; een tekenreeks met het pad van het JSFL-script dat wordt uitgevoerd, uitgedrukt als bestand:/// URI.
<code>fl.sourcePath</code>	Een tekenreeks die een lijst met items bevat in het globale pad voor de bron van ActionScript 3.0. Hiermee wordt de locatie van ActionScript-klassenbestanden aangegeven.
<code>fl.swfPanels</code>	Een array met geregistreerde swfPanel-objecten (zie <a href="#">Object swfPanel</a> ).
<code>fl.tools</code>	Alleen-lezen; een array met objecten Tools.
<code>fl.version</code>	Alleen-lezen; de lange-tekenreeksversie van het ontwerpgereedschap van Flash, inclusief platform.
<code>fl.xmlui</code>	Alleen-lezen; een <a href="#">Object XMLUI</a> .

## fl.actionsPanel

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.actionsPanel
```

### Beschrijving

Alleen-lezen eigenschap; een object actionsPanel, dat staat voor het nu weergegeven deelvenster Handelingen. Zie [Object actionsPanel](#) voor informatie over het gebruik van deze eigenschap.

## fl.addEventListener()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.addEventListener(eventType, callbackFunction)
```

### Parameters

**eventType** Een tekenreeks die het type gebeurtenis aanduidt dat aan deze callback-functie moet worden doorgegeven. Acceptabele waarden zijn "documentNew", "documentOpened", "documentClosed", "mouseMove", "documentChanged", "layerChanged" en "frameChanged".

De waarde `documentChanged` betekent niet dat de inhoud van een document is gewijzigd, maar dat er nu een ander document op de voorgrond staat. Dat wil zeggen dat `fl.getDocumentDOM()` een andere waarde retourneert dan vóór deze gebeurtenis.

**callbackFunction** De naam van de functie die u wilt uitvoeren zodra de gebeurtenis zich voordoet.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een functie geregistreerd die moet worden aangeroepen wanneer een bepaalde gebeurtenis zich voordoet.

Houd er bij deze methode rekening mee dat als de gebeurtenis zich vaak voordoet (bijvoorbeeld `mouseMove`) en de functie veel tijd nodig heeft voor de uitvoering, uw toepassing mogelijk kan vastlopen of anderszins in een foutstatus belandt.

### Voorbeeld

In het volgende voorbeeld wordt een bericht weergegeven in het deelvenster Uitvoer wanneer een document wordt gesloten:

```
myFunction = function () {  
    fl.trace('document was closed'); }  
fl.addEventListener("documentClosed", myFunction);
```

### Zie ook

[fl.removeEventListener\(\)](#)

## fl.as3PackagePaths

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`fl.as3PackagePaths`

### Beschrijving

Eigenschap; een tekenreeks die overeenkomt met de algemene instelling Classpath in het dialoogvenster ActionScript 3.0-instellingen. Items in de tekenreeks worden gescheiden door puntkomma's. Als u ActionScript 2.0 Classpath-instellingen wilt weergeven of wijzigen, gebruikt u [fl.packagePaths](#).

### Voorbeeld

In het volgende voorbeeld ziet u hoe u de ActionScript 3.0 Classpath-instellingen wijzigt.

```
fl.trace(fl.as3PackagePaths);  
// Output (assuming started with default value)  
// .;$(AppConfig)/ActionScript 3.0/Classes  
fl.as3PackagePaths="buying;selling";  
fl.trace(fl.as3PackagePaths);  
// Output  
// buying; selling
```

#### Zie ook

[fl.resetAS3PackagePaths\(\)](#)

## fl.browseForFileURL()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
fl.browseForFileURL(browseType [, title [, previewArea]])
```

#### Parameters

**browseType** Een tekenreeks die het type bladerbewerking in bestanden aangeeft. Acceptabele waarden zijn "open", "select" en "save". Met de waarden "open" en "select" opent u het systeemvenster Bestand openen. Elke waarde is beschikbaar omwille van compatibiliteit met Dreamweaver. Met de waarde "save" opent u het systeemvenster Bestand opslaan.

**title** Een tekenreeks die de titel voor het dialoogvenster Bestand openen of Bestand opslaan aangeeft. Wanneer deze parameter wordt weggelaten, wordt een standaardwaarde gebruikt. Deze parameter is optioneel.

**previewArea** Een optionele parameter die door Flash en Fireworks wordt genegeerd en alleen aanwezig is omwille van de compatibiliteit met Dreamweaver.

#### Geretourneerde waarde

De URL van het bestand, uitgedrukt als bestand:/// URI; retourneert null als de gebruiker het dialoogvenster annuleert.

#### Beschrijving

Methode; hiermee wordt het systeemvenster Bestand openen of Bestand opslaan geopend waarin de gebruiker een bestand kan opgeven dat moet worden geopend of opgeslagen.

#### Voorbeeld

In het volgende voorbeeld kan de gebruiker een FLA-bestand kiezen dat hij/zij wil openen. Vervolgens wordt het bestand geopend. (De methode `fl.browseForFileURL()` kan bladeren naar elk type bestand, maar `fl.openDocument()` kan alleen FLA-bestanden openen.)

```
var fileURL = fl.browseForFileURL("open", "Select file");  
var doc = fl.openDocument(fileURL);
```

#### Zie ook

[fl.browseForFolderURL\(\)](#)

## fl.browseForFolderURL()

### Beschikbaarheid

Flash 8.

### Gebruik

```
fl.browseForFolderURL([description])
```

### Parameters

**description** Een optionele tekenreeks die de beschrijving opgeeft van het dialoogvenster Bladeren naar map. Als deze parameter wordt weggelaten, wordt in het beschrijvingsgebied niets weergegeven.

### Geretourneerde waarde

De URL van de map, uitgedrukt als bestand:/// URI; retourneert `null` als de gebruiker het dialoogvenster annuleert.

### Beschrijving

Methode; hiermee wordt het dialoogvenster Bladeren naar map geopend waarin de gebruiker een map kan selecteren.

**Opmerking:** De titel van het dialoogvenster is altijd *Bladeren naar map*. Gebruik de parameter *description* om meer details aan het beschrijvingsgebied onder de titel toe te voegen, zoals “Selecteer een map” of “Selecteer het pad met het profiel dat u wilt importeren”.

### Voorbeeld

In het volgende voorbeeld kan de gebruiker een map selecteren en wordt vervolgens een lijst met bestanden in die map weergegeven:

```
var folderURI = fl.browseForFolderURL("Select a folder.");  
var folderContents = FLfile.listFolder(folderURI);
```

### Zie ook

[fl.browseForFileURL\(\)](#), [Object FLfile](#)

## fl.clearPublishCache()

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.clearPublishCache()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de publicatiecache geleegd.

### Voorbeeld

Met de volgende code wordt de publicatiecache geleegd:

```
fl.clearPublishCache()
```

### Zie ook

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), "[fl.publishCacheMemoryEntrySizeLimit](#)" op pagina 255, "[fl.publishCacheMemorySizeMax](#)" op pagina 255

## fl.clipCopyString()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.clipCopyString(string)
```

### Parameters

**tekenreeks** Een tekenreeks die naar het klembord moet worden gekopieerd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de opgegeven tekenreeks naar het klembord gekopieerd.

Als u de huidige selectie naar het klembord wilt kopiëren, gebruikt u [document.clipCopy\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt het pad van het huidige document naar het klembord gekopieerd:

```
var documentPath = fl.getDocumentDOM().path;  
fl.clipCopyString(documentPath);
```

## fl.closeAll()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.closeAll([bPromptToSave])
```

### Parameters

**bPromptToSave** Een optionele Booleaanse waarde die aangeeft of het dialoogvenster Opslaan moet worden weergegeven voor bestanden die zijn gewijzigd sinds de vorige keer dat deze zijn opgeslagen, of het dialoogvenster Opslaan als voor bestanden die nog niet eerder zijn opgeslagen. De standaardwaarde is `true`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden all geopende bestanden gesloten (FLA-bestanden, SWF-bestanden, JSFL-bestanden enzovoort). Als u alle geopende bestanden wilt sluiten zonder wijzigingen op te slaan, geeft u `false` door voor `bPromptToSave`. Met deze methode wordt de toepassing niet beëindigd.

### Voorbeeld

Met de volgende code worden alle geopende bestanden gesloten, en wordt de gebruiker gevraagd nieuwe of gewijzigde bestanden op te slaan.

```
fl.closeAll();
```

### Zie ook

[fl.closeAllPlayerDocuments\(\)](#), [fl.closeDocument\(\)](#)

## fl.closeAllPlayerDocuments()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.closeAllPlayerDocuments();
```

### Parameters

Geen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als een of meer filmvensters geopend waren, anders `false`.

### Beschrijving

Methode; hiermee worden alle SWF-bestanden gesloten die zijn geopend met Besturing > Film testen.

### Voorbeeld

In het volgende voorbeeld worden alle SWF-bestanden gesloten die zijn geopend met Besturing > Film testen.

```
fl.closeAllPlayerDocuments();
```

### Zie ook

[fl.closeAll\(\)](#), [fl.closeDocument\(\)](#)

## fl.closeDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.closeDocument (documentObject [, bPromptToSaveChanges])
```

### Parameters

**documentObject** Een [Object Document](#). Als *documentObject* naar het actieve document verwijst, wordt het venster Document mogelijk pas gesloten wanneer het script dat deze methode aanroept, volledig is uitgevoerd.

**bPromptToSaveChanges** Een Booleaanse waarde. Wanneer *bPromptToSaveChanges* `false` is, wordt de gebruiker niet gevraagd of het document onopgeslagen wijzigingen bevat. Het bestand wordt dan gesloten en de wijzigingen genegeerd. Als *bPromptToSaveChanges* `true` is en als het document onopgeslagen wijzigingen bevat, wordt het standaardvenster weergegeven met de vraag of de gebruiker het bestand wil opslaan. De standaardwaarde is `true`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het opgegeven document gesloten.

### Voorbeeld

In het volgende voorbeeld ziet u twee manieren om een document te sluiten.

```
// Closes the specified document and prompts to save changes.
fl.closeDocument (fl.documents[0]);
fl.closeDocument (fl.documents[0] , true); // Use of true is optional.
// Closes the specified document without prompting to save changes.
fl.closeDocument (fl.documents[0] , false);
```

### Zie ook

[fl.closeAll\(\)](#)

## fl.compilerErrors

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.compilerErrors
```

### Beschrijving

Alleen-lezen eigenschap; een object `compilerErrors` dat staat voor het deelvenster Fouten. Zie [Object compilerErrors](#) voor informatie over het gebruik van deze eigenschap.

## fl.componentsPanel

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.componentsPanel
```

### Beschrijving

Alleen-lezen eigenschap; een [Object componentsPanel](#) dat staat voor het deelvenster Componenten.

### Voorbeeld

@@@In het volgende voorbeeld wordt een object componentsPanel opgeslagen in de variabele comPanel:

```
var comPanel = fl.componentsPanel;
```

## fl.configDirectory

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.configDirectory
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het volledige pad voor de map Configuration van de lokale gebruiker in een platformspecifieke notatie aanduidt. Als u dit pad wilt opgeven als een bestand:/// URI (niet platformspecifiek), gebruikt u [fl.configURI](#).

### Voorbeeld

In het volgende voorbeeld wordt de map Configuration weergegeven in het deelvenster Uitvoer:

```
fl.trace("My local configuration directory is " + fl.configDirectory);
```

## fl.configURI

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.configURI
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het volledige pad voor de map Configuration van de lokale gebruiker als een bestand:/// URI aanduidt. Zie ook [fl.configDirectory](#).



### Voorbeeld

In het volgende voorbeeld wordt een opgegeven script uitgevoerd. Met `fl.configURI` kunt u de locatie van het script opgeven zonder te weten op welk platform het script wordt uitgevoerd.

```
// To run a command in your commands menu, change "Test.jsfl"  
// to the command you want to run in the line below.  
fl.runScript( fl.configURI + "Commands/Test.jsfl" );
```

## fl.contactSensitiveSelection

### Beschikbaarheid

Flash 8.

### Gebruik

```
fl.contactSensitiveSelection
```

### Beschrijving

Een Booleaanse waarde die opgeeft of de selectiemodus Contactgevoelig is ingeschakeld (`true`) of niet (`false`).

### Voorbeeld

In het volgende voorbeeld ziet u hoe u de selectiemodus Contactgevoelig uitschakelt voordat u een selectie maakt en hoe u deze op de oorspronkelijke waarde terugzet nadat u een selectie hebt gemaakt:

```
var contact = fl.contactSensitiveSelection;  
fl.contactSensitiveSelection = false;  
// Insert selection code here.  
fl.contactSensitiveSelection = contact;
```

## fl.createDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.createDocument ( [docType] )
```

### Parameters

**docType** Een tekenreeks die het type document aangeeft dat moet worden gemaakt. Acceptabele waarden zijn "timeline", "presentation" en "application". De standaardwaarde is "timeline", wat hetzelfde effect heeft als het kiezen van Bestand > Nieuw > Flash-bestand (ActionScript 3.0). Deze parameter is optioneel.

### Geretourneerde waarde

Het object Document voor het nieuwe document, als de methode succesvol is. Als er een fout optreedt, is de waarde `undefined`.

### Beschrijving

Methode; hiermee wordt een nieuw document geopend en geselecteerd. Waarden voor grootte, resolutie en kleur zijn gelijk aan de huidige standaardwaarden.

### Voorbeeld

In het volgende voorbeeld worden verschillende typen documenten gemaakt:

```
// Create two Timeline-based Flash documents.  
fl.createDocument();  
fl.createDocument("timeline");  
// Create a Slide Presentation document.  
fl.createDocument("presentation");  
// Create a Form Application document.  
fl.createDocument("application");
```

## fl.createNewDocList

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.createNewDocList
```

### Beschrijving

Alleen-lezen eigenschap; een array van tekenreeksen die staan voor de verschillende typen documenten die kunnen worden gemaakt.

### Voorbeeld

In het volgende voorbeeld worden de typen documenten weergegeven die in het deelvenster Uitvoer kunnen worden gemaakt:

```
fl.trace("Number of choices " + fl.createNewDocList.length);  
for (i = 0; i < fl.createNewDocList.length; i++)  
    fl.trace("choice: " + fl.createNewDocList[i]);
```

## fl.createNewDocListType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.createNewDocListType
```

### Beschrijving

Alleen-lezen eigenschap; een array van tekenreeksen die staan voor de bestandsextensies van de typen documenten die kunnen worden gemaakt. De gegevens in de array komen direct overeen (per index) met de gegevens in de array [fl.createNewDocList](#).

### Voorbeeld

In het volgende voorbeeld worden de extensies van de typen documenten weergegeven die in het deelvenster Uitvoer kunnen worden gemaakt:

```
fl.trace("Number of types " + fl.createNewDocListType.length);  
for (i = 0; i < fl.createNewDocListType.length; i++) fl.trace("type: " +  
fl.createNewDocListType[i]);
```

## fl.createNewTemplateList

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.createNewTemplateList
```

### Beschrijving

Alleen-lezen eigenschap; een array van tekenreeksen die staan voor de verschillende typen sjablonen die kunnen worden gemaakt.

### Voorbeeld

In het volgende voorbeeld worden de typen sjablonen weergegeven die in het deelvenster Uitvoer kunnen worden gemaakt:

```
fl.trace("Number of template types: " + fl.createNewTemplateList.length); for (i = 0; i <  
fl.createNewTemplateList.length; i++) fl.trace("type: " + fl.createNewTemplateList[i]);
```

## fl.documents

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.documents
```

### Beschrijving

Alleen-lezen eigenschap; een array van objecten Document (zie [Object Document](#)) die staan voor de documenten (FLA-bestanden) die momenteel geopend zijn en kunnen worden bewerkt.

### Voorbeeld

In het volgende voorbeeld wordt een array van geopende documenten opgeslagen in de variabele docs:

```
var docs = fl.documents;
```

In het volgende voorbeeld worden de namen van momenteel geopende documenten in het deelvenster Uitvoer weergegeven:

```
for (doc in fl.documents) {  
    fl.trace(fl.documents[doc].name);  
}
```

## fl.drawingLayer

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`fl.drawingLayer`

### Beschrijving

Alleen-lezen eigenschap; het [Object drawingLayer](#) dat door een uitbreidbaar gereedschap moet worden gebruikt wanneer de gebruiker tijdelijk wil tekenen tijdens het slepen (bijvoorbeeld tijdens het maken van een selectiekader).

### Voorbeeld

Zie [drawingLayer.setColor\(\)](#).

## fl.exportPublishProfileString()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
fl.exportPublishProfileString( ucFURI [, profileName] )
```

### Parameters

**ucFURI** Een tekenreeks die de URI (Uniform Resource Identifier) van het bestand opgeeft van waaruit de publicatie-instellingen geëxporteerd moeten worden.

**profileName** Een tekenreeks die de te exporteren profielnaam opgeeft. Deze parameter is optioneel.

### Geretourneerde waarde

Tekenreeks.

### Beschrijving

Hiermee wordt het publicatieprofiel van een specifiek document geretourneerd zonder dat u het bestand hoeft te openen. U kunt desgewenst ook het publicatieprofiel opgeven.

### Voorbeeld

In het volgende voorbeeld wordt de tekenreeks voor het publicatieprofiel gelezen:

**Object flash (fl)**

```
var ppXML = "";
var ucfURI = fl.browseForFileURL("open", "select a FLA");
if (ucfURI && ucfURI.length > 0)
ppXML = fl.exportPublishProfileString(ucfURI);
fl.trace(ppXML);
```

## fl.externalLibraryPath

**Beschikbaarheid**

Flash CS4 Professional.

**Gebruik**

```
fl.externalLibraryPath
```

**Beschrijving**

Eigenschap; een tekenreeks die een lijst met items bevat in het globale pad voor de externe bibliotheek van ActionScript 3.0. Hiermee wordt de locatie van SWC-bestanden aangegeven die worden gebruikt als gedeelde runtime bibliotheken. Items in de tekenreeks worden gescheiden door puntkomma's. In het ontwerp gereedschap worden de items opgegeven door Bewerken > Voorkeuren > ActionScript > ActionScript 3.0-instellingen te kiezen.

**Voorbeeld**

In het volgende voorbeeld wordt de map /SWC\_runtime toegevoegd aan het globale pad naar de externe bibliotheek van ActionScript 3.0:

```
fl.trace(fl.externalLibraryPath);
fl.externalLibraryPath = "/SWC_runtime;" + fl.externalLibraryPath;
fl.trace(fl.externalLibraryPath);
```

**Zie ook**

[fl.flexSDKPath](#), [fl.libraryPath](#), [fl.sourcePath](#), [document.externalLibraryPath](#)

## fl.fileExists()

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
fl.fileExists(fileURI)
```

**Parameters**

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, dat het pad naar het bestand aangeeft.

**Geretoureerde waarde**

Een Booleaanse waarde: `true` als het bestand op schijf aanwezig is, anders `false`.

### Beschrijving

Methode; hiermee wordt gecontroleerd of een bestand al op schijf aanwezig is.

### Voorbeeld

In het volgende voorbeeld wordt voor elk opgegeven bestand `true` of `false` weergegeven in het deelvenster Uitvoer, afhankelijk van de vraag of het bestand aanwezig is.

```
alert(fl.fileExists("file:///C:/example fla"));  
alert(fl.fileExists("file:///C:/example.jsfl"));  
alert(fl.fileExists(""));
```

## fl.findDocumentDOM()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.findDocumentDOM(id)
```

### Parameters

**id** Een geheel getal dat een unieke id voor een document aangeeft.

### Geretourneerde waarde

Een object Document, of `null` als er geen document bestaat met de opgegeven *id*.

### Beschrijving

Methode; hiermee kunt u een specifiek bestand zoeken aan de hand van de unieke id (in plaats van bijvoorbeeld de indexwaarde). Gebruik deze methode in combinatie met `document.id`.

### Voorbeeld

In het volgende voorbeeld ziet u hoe de id van een document wordt gelezen en vervolgens wordt gebruikt om dat document te zoeken:

```
var originalDocID = fl.getDocumentDOM().id;  
// other code here, maybe working in different files  
var targetDoc = fl.findDocumentDOM(originalDocID);  
// Set the height of the Stage in the original document to 400 pixels.  
targetDoc.height = 400;
```

### Zie ook

[fl.findDocumentIndex\(\)](#)

## fl.findDocumentIndex()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.findDocumentIndex(name)
```

### Parameters

**name** De documentnaam waarvoor u de index wilt vinden. Het document moet geopend zijn.

### Geretourneerde waarde

Een array van gehele getallen die staan voor de positie *name* in de array `fl.documents`.

### Beschrijving

Methode; hiermee wordt een array geretourneerd van gehele getallen die staan voor de positie van *name* in de array `fl.documents`. Er kunnen meerdere documenten met dezelfde naam geopend zijn (als de documenten in verschillende mappen staan).

### Voorbeeld

In het volgende voorbeeld wordt informatie in het deelvenster Uitvoer weergegeven over de indexpositie van geopende bestanden met de naam test.fla:

```
var filename = "test.fla"
var docIndex = fl.findDocumentIndex(filename);
for (var index in docIndex)
    fl.trace(filename + " is open at index " + docIndex[index]);
```

### Zie ook

[fl.documents](#), [fl.findDocumentDOM\(\)](#)

## fl.findObjectInDocByName()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.findObjectInDocByName(instanceName, document)
```

### Parameters

**instanceName** Een tekenreeks met de instantienaam van een item in het opgegeven document.

**document** Het [Object Document](#) waarin moet worden gezocht naar het opgegeven item.

### Geretourneerde waarde

Een array van algemene objecten. Gebruik de eigenschap `.obj` van elk item in de array om het object op te halen. Het object heeft de volgende eigenschappen: `keyframe`, `layer`, `timeline` en `parent`. U kunt deze eigenschappen gebruiken om toegang te krijgen tot de hiërarchie van het object. Zie [fl.findObjectInDocByType\(\)](#) voor meer informatie over deze eigenschappen en hoe u er toegang toe kunt krijgen.

U kunt ook toegang krijgen tot methoden en eigenschappen voor de waarden `layer` en `timeline`. Deze zijn respectievelijk gelijk aan het [Object Layer](#) en het [Object Timeline](#).

### Beschrijving

Methode; hiermee wordt gezocht naar elementen in een document met instantienamen die met de opgegeven tekst overeenkomen.

**Opmerking:** *In sommige gevallen werkt deze methode alleen wanneer deze vanuit een FLA-bestand als een opdracht wordt uitgevoerd, niet wanneer u het JSFL-bestand bekijkt of bewerkt.*

### Voorbeeld

In het volgende voorbeeld wordt in het huidige document gezocht naar elementen met de naam "instance01".

```
var nameToSearchFor = "instance01";
var doc = fl.getDocumentDOM();
var results = fl.findObjectInDocByName(nameToSearchFor, doc);
if (results.length > 0) {
    alert("success, found " + results.length + " objects");
}
else {
    alert("failed, no objects named " + nameToSearchFor + " found");
}
```

### Zie ook

[fl.findObjectInDocByType\(\)](#)

## fl.findObjectInDocByType()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.findObjectInDocByType(elementType, document)
```

### Parameters

**elementType** Een tekenreeks die staat voor het type element waarnaar u wilt zoeken. Zie [element.elementType](#) voor geldige waarden.

**document** Het [Object Document](#) waarin moet worden gezocht naar het opgegeven item.

### Geretoureerde waarde

Een array van algemene objecten. Gebruik de eigenschap `.obj` van elk item in de array om het elementobject op te halen. Elk object heeft de volgende eigenschappen: `keyframe`, `layer`, `timeline` en `parent`. U kunt deze eigenschappen gebruiken om toegang te krijgen tot de hiërarchie van het object.

U kunt ook toegang krijgen tot methoden en eigenschappen voor de waarden `layer` en `timeline`. Deze zijn respectievelijk gelijk aan het [Object Layer](#) en het [Object Timeline](#).

In het tweede en derde voorbeeld in de sectie Voorbeelden hieronder ziet u hoe u deze eigenschappen kunt gebruiken.

### Beschrijving

Methode; hiermee wordt gezocht naar elementen van een specifiek elementtype in een document.



**Opmerking:** In sommige gevallen werkt deze methode alleen wanneer deze vanuit een FLA-bestand als een opdracht wordt uitgevoerd, niet wanneer u het JSFL-bestand bekijkt of bewerkt.

### Voorbeeld

In het volgende voorbeeld wordt in het huidige document gezocht naar tekstvelden, waarna de inhoud ervan wordt gewijzigd:

```
var doc = fl.getDocumentDOM();
var typeToSearchFor = "text";
var results = fl.findObjectInDocByType(typeToSearchFor, doc);
if (results.length > 0) {
    for (var i = 0; i < results.length; i++) {
        results[i].obj.setTextString("new text");
    }
    alert("success, found " + results.length + " objects");
}
else {
    alert("failed, no objects of type " + typeToSearchFor + " found");
}
```

In het volgende voorbeeld ziet u hoe u de speciale eigenschappen van het object kunt gebruiken die door deze methode worden geretourneerd:

```
var doc = fl.getDocumentDOM();
var resultsArray = findObjectInDocByType("text", doc);
if (resultsArray.length > 0)
{
    var firstItem = resultsArray[0];

    // firstItem.obj- This is the element object that was found.

    // You can access the following properties of this object:
    // firstItem.keyframe- The keyframe that the element is on.
    // firstItem.layer- The layer that the keyframe is on.
    // firstItem.timeline- The timeline that the layer is on.
    // firstItem.parent- The parent of the timeline. For example,
    // the timeline might be in a symbol instance.
}
```

In het volgende voorbeeld ziet u hoe u een back-up van de DOM kunt maken om de naam van een laag te zoeken waarop een tekstveld is gevonden, met behulp van het object resultArray.obj:

```
var doc = fl.getDocumentDOM();
var typeToSearchFor = "text";
var resultsArray = fl.findObjectInDocByType(typeToSearchFor, doc);
if (resultsArray.length > 0) {
    for (var i = 0; i < resultsArray.length; i++) {
        resultsArray[i].obj.setTextString("new text");
        var firstItem = resultsArray[0];
        firstItemObj = firstItem.obj;
        fl.trace(firstItemObj.layer.name+"layerName");
    }
}
else {
    alert("failed, no objects of type " + typeToSearchFor + " found");
}
```

### Zie ook

[fl.findObjectInDocByName\(\)](#)

## fl.flexSDKPath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fl.flexSDKPath
```

### Beschrijving

Eigenschap; een tekenreeks die het pad naar de Flex-map SDK opgeeft; deze map bevat de map bin, frameworks, lib en andere mappen. In het ontwerpgeredeenschap worden de items opgegeven door Bewerken > Voorkeuren > ActionScript > ActionScript 3.0-instellingen te kiezen.

### Voorbeeld

In het volgende voorbeeld wordt het Flex SDK-pad in het deelvenster Uitvoer weergegeven:

```
fl.trace(fl.flexSDKPath);
```

### Zie ook

[fl.externalLibraryPath](#), [fl.libraryPath](#), [fl.sourcePath](#)

## fl.getAppMemoryInfo()

### Beschikbaarheid

Flash 8 (alleen Windows).

### Gebruik

```
fl.getAppMemoryInfo(memType)
```

### Parameters

**memType** Een geheel getal dat het geheugengebied aangeeft waarop een query moet worden uitgevoerd. Zie de volgende beschrijving voor een lijst met geldige waarden.

### Geretourneerde waarde

Een geheel getal dat staat voor het aantal bytes dat in een bepaald gebied van het Flash.exe-geheugen wordt gebruikt.

### Beschrijving

Methode (alleen Windows); hiermee wordt een geheel getal geretourneerd dat staat voor het aantal bytes dat in een bepaald gebied van het Flash.exe-geheugen wordt gebruikt. Aan de hand van de volgende tabel kunt u vaststellen welke waarde u wilt doorgeven als *memType*:

memType	Brongegevens
0	PAGEFAULTCOUNT
1	PEAKWORKINGSETSIZE
2	WORKINGSETSIZE
3	QUOTAPEAKPAGEDPOOLUSAGE
4	QUOTAPAGEDPOOLUSAGE
5	QUOTAPEAKNONPAGEDPOOLUSAGE
6	QUOTANONPAGEDPOOLUSAGE
7	PAGEFILEUSAGE
8	PEAKPAGEFILEUSAGE

### Voorbeeld

In het volgende voorbeeld wordt het huidige werkgeheugengebruik weergegeven:

```
var memsize = fl.getAppMemoryInfo(2);  
fl.trace("Flash current memory consumption is " + memsize + " bytes or " + memsize/1024 + "  
KB");
```

## fl.getDocumentDOM()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.getDocumentDOM()
```

### Parameters

Geen.

### Geretourneerde waarde

Een object Document of null als er geen documenten geopend zijn.

### Beschrijving

Methode; hiermee wordt de DOM ([Object Document](#)) opgehaald van het actieve document (FLA-bestand). Als er een of meer documenten geopend zijn maar een document niet de focus heeft (bijvoorbeeld als een JSFL-bestand de focus heeft), wordt met deze methode de DOM opgehaald van het document dat het laatst actief is geweest.

### Voorbeeld

In het volgende voorbeeld wordt in het deelvenster Uitvoer de naam weergegeven van het document dat actief is of als laatste actief was:

```
var currentDoc = fl.getDocumentDOM();  
fl.trace(currentDoc.name);
```

## fl.getSwfPanel()

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.getSwfPanel(panelName, [useLocalizedPanelName])
```

### Parameters

**panelName** De gelokaliseerde deelvenster naam of hoofbestandsnaam van het SWF-bestand van het deelvenster. Geef false door als de tweede parameter als u het SWF-bestand gebruikt.

**useLocalizedPanelName** Optioneel. Standaardwaarde is true. In geval van false wordt ervan uitgegaan dat de parameter panelName de Engelse (niet vertaalde) naam van het deelvenster is. Deze naam komt overeen met de naam van het SWF-bestand zonder de bestandstoevoeging.

### Geretourneerde waarde

Het object SWFPanel.

### Beschrijving

Methode; retourneert het SWFPanel-object op basis van de gelokaliseerde naam van het deelvenster of de SWF-bestandsnaam van het deelvenster (zonder de bestandsnaamtoevoeging).

### Voorbeeld

In het volgende voorbeeld wordt de naam van het deelvenster weergegeven waarnaar in het deelvenster Uitvoer wordt verwezen als 'Project':

```
fl.trace('name of panel is: ' + fl.getSwfPanel('Project').name);
```

## fl.installedPlayers

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.installedPlayers()
```

### Parameters

Geen.

### Geretourneerde waarde

Een array van generieke objecten die overeenkomen met de lijst geïnstalleerde Flash Players in de Eigenschapcontrole van het document.

### Beschrijving

Alleen-lezeneigenschap; de array van generieke objecten die overeenkomen met de lijst geïnstalleerde Flash Players in de Eigenschapcontrole van het document.

Elk object in de array bevat de volgende eigenschappen:

**name** De tekenreeks met de naam van het document.

**versie** Hiermee kan de huidige speler worden ingesteld voor een document aan de hand van de functie `Document.setPlayerVersion()`.

**minASVersion** De minimaal door het document vereiste ActionScript-versie. Een geheel getal tussen `minASVersion` en `maxASVersion` (inclusief) kan worden gebruikt om de ActionScript-versie van het document in te stellen aan de hand van de eigenschap `Document.asVersion`.

**maxASVersion** De hoogste ActionScript-versie die door het document wordt ondersteund.

**stageWidth** De standaardbreedte van het werkgebied in pixels voor het opgegeven doel. Voor iPhone is het standaardformaat bijvoorbeeld 320 x 480 pixels. Voor Android is het standaardformaat 480 x 800 pixels.

**stageHeight** De standaardhoogte van het werkgebied in pixels voor het opgegeven doel. Voor iPhone is het standaardformaat bijvoorbeeld 320 x 480 pixels. Voor Android is het standaardformaat 480 x 800 pixels.

### Voorbeeld

In het volgende voorbeeld worden de eigenschappen van alle objecten in de `installedPlayers`-array getraceerd naar het uitvoervenster:

```
var arr = fl.installedPlayers;
for (var i in arr) fl.trace("name: " + arr[i].name + " version: " + arr[i].version + "
minASVersion: " + arr[i].minASVersion + " maxASVersion: " + arr[i].maxASVersion + " stageWidth:
" + arr[i].stageWidth + " stageHeight: " + arr[i].stageHeight + " ");
```

## fl.isFontInstalled()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fl.isFontInstalled(fontName)
```

### Parameters

**fontName** Een tekenreeks die de naam van een apparaatlettertype aangeeft.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het opgegeven lettertype is geïnstalleerd, anders `false`.

### Beschrijving

Methode: hiermee wordt bepaald of een specifiek lettertype is geïnstalleerd.

### Voorbeeld

Met de volgende code wordt “true” weergegeven in het deelvenster Uitvoer als het lettertype Times is geïnstalleerd

```
fl.trace(fl.isFontInstalled("Times"));
```

## fl.languageCode

### Beschikbaarheid

Flash CS5 Professional.

### Gebruik

`fl.languageCode`

### Beschrijving

Eigenschap, een tekenreeks waarmee de code van vijf tekens wordt geretourneerd die de locale aangeeft van de gebruikersinterface van de toepassing.

### Voorbeeld

Het volgende voorbeeld retourneert de taalcode van vijf tekens die wordt aangegeven door de gelocaliseerde gebruikersinterface van de Flash-toepassing:

```
locConfigURI = fl.applicationURI + fl.languageCode + "/Configuration";
```

## fl.libraryPath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`fl.libraryPath`

### Beschrijving

Eigenschap; een tekenreeks die een lijst met items bevat in het globale pad voor de bibliotheek van ActionScript 3.0. Hiermee wordt de locatie aangegeven van SWC-bestanden of mappen die SWC-bestanden bevatten. Items in de tekenreeks worden gescheiden door puntkomma's. In het ontwerpgereedschap worden de items opgegeven door Bewerken > Voorkeuren > ActionScript > ActionScript 3.0-instellingen te kiezen.

### Voorbeeld

In het volgende voorbeeld wordt de map /SWC toegevoegd aan het globale pad naar de bibliotheek van ActionScript 3.0:

```
fl.trace(fl.libraryPath);  
fl.libraryPath = "/SWC;" + fl.libraryPath;  
fl.trace(fl.libraryPath);
```

### Zie ook

[fl.externalLibraryPath](#), [fl.flexSDKPath](#), [fl.sourcePath](#), [document.libraryPath](#)

## fl.mapPlayerURL()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.mapPlayerURL(URI [, returnMBCS])
```

### Parameters

**URI** Een tekenreeks die de Unicode-URL met escape-teken bevat die moet worden toegewezen.

**returnMBCS** Een Booleaanse waarde. Stel `true` in als u een MBCS-pad met escape-teken als retourwaarde wilt. Anders retourneert de methode UTF-8. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Een tekenreeks die de omgezette URL is.

### Beschrijving

Methode; hiermee wordt een Unicode URL met escape-teken toegewezen aan een UTF-8 of MBCS URL. Gebruik deze methode wanneer de tekenreeks in ActionScript wordt gebruikt om toegang te krijgen tot een externe bron. U moet deze methode gebruiken als u multibyte-teken moet verwerken.

### Voorbeeld

In het volgende voorbeeld wordt een URL omgezet in UTF-8 zodat de speler deze kan laden:

```
var url = MMExecute( "fl.mapPlayerURL(" + myURL + ", false);" );  
mc.loadMovie( url );
```

## fl.Math

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.Math
```

### Beschrijving

Alleen-lezen eigenschap; het [Object Math](#) biedt methoden voor matrix- en puntbewerkingen.

### Voorbeeld

In het volgende voorbeeld ziet u de transformatiematrix van het geselecteerde object en de inverse ervan:

**Object flash (fl)**

```
// Select an element on the Stage and then run this script.
var mat =fl.getDocumentDOM().selection[0].matrix;
for(var prop in mat){
fl.trace("mat."+prop+" = " + mat[prop]);
}
var invMat = fl.Math.invertMatrix( mat );
for(var prop in invMat) {
fl.trace("invMat."+prop+" = " + invMat[prop]);
}
```

## fl.mruRecentFileList

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
fl.mruRecentFileList
```

**Beschrijving**

Alleen-lezen eigenschap; een array van de volledige bestandsnamen in de MRU-lijst (Most Recently Used, laatst gebruikt) die in het Flash-ontwerpgereedschap wordt beheerd.

**Voorbeeld**

In het volgende voorbeeld wordt het aantal onlangs geopende bestanden in het deelvenster Uitvoer weergegeven, plus de naam van elk bestand:

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileList.length);
for (i = 0; i < fl.mruRecentFileList.length; i++) fl.trace("file: " + fl.mruRecentFileList[i]);
```

## fl.mruRecentFileListType

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
fl.mruRecentFileListType
```

**Beschrijving**

Alleen-lezen eigenschap; een array van de bestandstypen in de MRU-lijst die in het Flash-ontwerpgereedschap wordt beheerd. Deze array komt overeen met de array in de eigenschap [fl.mruRecentFileList](#).

**Voorbeeld**

In het volgende voorbeeld wordt het aantal onlangs geopende bestanden in het deelvenster Uitvoer weergegeven, plus het type van elk bestand:

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileListType.length);
for (i = 0; i < fl.mruRecentFileListType.length; i++) fl.trace("type: " +
fl.mruRecentFileListType[i]);
```



## fl.objectDrawingMode

### Beschikbaarheid

Flash 8.

### Gebruik

```
fl.objectDrawingMode
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of de objecttekenmodus is ingeschakeld (`true`) of dat de samenvoegtekenmodus is ingeschakeld (`false`).

### Voorbeeld

In het volgende voorbeeld wordt de status van de objecttekenmodus afgewisseld:

```
var toggleMode = fl.objectDrawingMode;
if (toggleMode) {
    fl.objectDrawingMode = false;
} else {
    fl.objectDrawingMode = true;
}
```

## fl.openDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.openDocument(fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de naam van het te openen bestand aangeeft.

### Geretoureerde waarde

Het [Object Document](#) voor het nieuwe document, als de methode succesvol is. Als het bestand niet wordt gevonden of geen geldig FLA-bestand is, wordt een fout gemeld en wordt het script geannuleerd.

### Beschrijving

Methode; hiermee wordt een Flash-document (FLA-bestand) geopend om te bewerken in een nieuw Flash-documentvenster. Het document krijgt de focus. Voor een gebruiker is het effect hetzelfde als het selecteren van Bestand > Openen en vervolgens het bestand selecteren. Als het opgegeven bestand al is geopend, komt het venster met het document naar de voorgrond. Het venster met het opgegeven bestand wordt het nu geselecteerde document.

### Voorbeeld

In het volgende voorbeeld wordt een bestand met de naam Document fla geopend dat is opgeslagen in de hoofdmap op station C. De code slaat een object Document op dat staat voor dat document in de variabele `doc` en stelt het document in als het geselecteerde document. Dat wil zeggen, dat `fl.getDocumentDOM()` naar dit document verwijst totdat de focus wordt gewijzigd.

```
var doc = fl.openDocument("file:///c:/Document fla");
```

## fl.openScript()

### Beschikbaarheid

Flash MX 2004. Optionele parameters die in Flash Professional CS5 zijn toegevoegd.

### Gebruik

```
fl.openScript(fileURI [, createExtension, className])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad van het JSFL-, AS-, ASC-, XML-, TXT-bestand of een ander bestand aangeeft dat in de teksteditor van Flash moet worden geladen. Deze parameter mag niet null zijn. Wanneer de waarde null is, wordt met deze methode een nieuw script geopend van het type dat is opgegeven door de parameter `createExtension`.

**createExtension** Een tekenreeks waarmee wordt bepaald welk type document wordt gemaakt als `fileURI` null is. De standaardwaarde is 'AS'; tot de toegestane waarden behoren 'JSFL', 'AS', 'ASC', 'XML', 'TXT', 'AS3\_CLASS' of 'AS3\_INTERFACE'. Deze parameter is toegevoegd in Flash Professional CS5.

**className** Een tekenreeks die wordt gebruikt om de volledig gekwalificeerde klassennaam op te geven als er een klasse of interface wordt gemaakt (zoals bepaald door de parameter `createExtension`). Deze parameter is toegevoegd in Flash Professional CS5.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode, hiermee wordt een bestaand bestand geopend of een nieuw script (.JSFL, .AS, .ASC) of een ander bestand (.XML, .TXT) gemaakt in de Flash-teksteditor.

### Voorbeeld

In het volgende voorbeeld wordt een bestand met de naam my\_test.jsfl geopend dat is opgeslagen in de map /temp op station C.

```
fl.openScript("file:///c:/temp/my_test.jsfl");
```

### Voorbeeld

In het volgende voorbeeld wordt een nieuw AS-bestand gemaakt met een lege AS3-klassedefinitie:

```
fl.openScript(null, 'AS3_CLASS');
```

## fl.outputPanel

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`fl.outputPanel`

### Beschrijving

Alleen-lezen eigenschap; verwijzing naar het [Object outputPanel](#).

### Voorbeeld

Zie [Object outputPanel](#).

## fl.packagePaths

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`fl.packagePaths`

### Beschrijving

Eigenschap; een tekenreeks die overeenkomt met de algemene instelling Classpath in het dialoogvenster ActionScript 2.0-instellingen. Klassepaden binnen de tekenreeks worden van elkaar gescheiden met een puntkomma (;). Als u ActionScript 3.0 Classpath-instellingen wilt weergeven of wijzigen, gebruikt u [fl.as3PackagePaths](#).

### Voorbeeld

In het volgende voorbeeld ziet u hoe u de ActionScript 2.0 Classpath-instellingen wijzigt:

```
fl.trace(fl.packagePaths);  
// Output (assuming started with default value)  
// .;$(LocalData)/Classes  
fl.packagePaths="buying;selling";  
fl.trace(fl.packagePaths);  
// Output  
// buying; selling
```

### Zie ook

[fl.resetPackagePaths\(\)](#)

## fl.presetPanel

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fl.presetPanel
```

### Beschrijving

Alleen-lezen eigenschap: een [Object presetPanel](#).

## fl.publishCacheDiskSizeMax

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.publishCacheDiskSizeMax
```

### Beschrijving

Eigenschap: een geheel getal dat de maximale grootte van de publicatiecache op de schijf instelt, uitgedrukt in megabytes.

### Voorbeeld

Met de volgende code wordt de maximale grootte van de publicatiecache op de schijf ingesteld op 1 MB:

```
fl.publishCacheDiskSizeMax = 1
```

### Zie ook

[fl.clearPublishCache\(\)](#), [fl.publishCacheEnabled](#), "[fl.publishCacheMemoryEntrySizeLimit](#)" op [pagina 255](#), "[fl.publishCacheMemorySizeMax](#)" op [pagina 255](#)

## fl.publishCacheEnabled

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.publishCacheEnabled
```

### Beschrijving

Eigenschap: een booleaanse waarde die instelt of de publicatiecache ingeschakeld is.

### Voorbeeld

De volgende code toont of de publicatiecache is ingeschakeld in het uitvoervenster.

```
fl.trace(fl.publishCacheEnabled);
```

### Zie ook

[fl.publishCacheDiskSizeMax](#), [fl.clearPublishCache\(\)](#), "[fl.publishCacheMemoryEntrySizeLimit](#)" op [pagina 255](#), "[fl.publishCacheMemorySizeMax](#)" op [pagina 255](#)

## fl.publishCacheMemoryEntrySizeLimit

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.publishCacheMemoryEntrySizeLimit
```

### Beschrijving

Eigenschap: een geheel getal dat de in kilobytes uitgedrukte maximale grootte instelt van vermeldingen die aan de publicatiecache in het geheugen kunnen worden toegevoegd. Even grote of kleinere vermeldingen blijven in het geheugen staan, grotere vermeldingen worden weggeschreven naar de schijf.

Gebruikers die over veel geheugen beschikken, kunnen deze waarde verhogen om de prestaties te verbeteren. Gebruikers die over weinig geheugen beschikken, kunnen deze waarde juist beter verlagen zodat de publicatiecache niet te veel geheugen verbruikt.

### Voorbeeld

Met de volgende code stelt u een maximale grootte van 100 kilobytes in voor vermeldingen in de publicatiecache die in het geheugen kunnen worden opgeslagen:

```
fl.publishCacheMemoryEntrySizeLimit = 100
```

### Zie ook

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), "[fl.clearPublishCache\(\)](#)" op pagina 230, "[fl.publishCacheMemorySizeMax](#)" op pagina 255

## fl.publishCacheMemorySizeMax

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
fl.publishCacheMemorySizeMax
```

### Beschrijving

Eigenschap: een geheel getal dat de maximale grootte van de publicatiecache in het geheugen instelt, uitgedrukt in megabytes.

### Voorbeeld

Met de volgende code wordt de maximale grootte van de publicatiecache in het geheugen ingesteld op 1 MB:

```
fl.publishCacheMemorySizeMax = 1
```

### Zie ook

[fl.publishCacheDiskSizeMax](#), [fl.publishCacheEnabled](#), "[fl.publishCacheMemoryEntrySizeLimit](#)" op pagina 255, "[fl.clearPublishCache\(\)](#)" op pagina 230

## fl.publishDocument()

### Beschikbaarheid

Flash CS5 Professional.

### Gebruik

```
fl.publishDocument( flaURI [, publishProfile] )
```

### Parameters

**flaURI** Een tekenreeks die wordt uitgedrukt als een bestand:/// URI die het pad van het FLA-bestand opgeeft dat automatisch moet worden gepubliceerd.

**publishProfile** Een tekenreeks die opgeeft welk publicatieprofiel moet worden gebruikt voor het publiceren. Als niets voor deze parameter wordt ingevuld, wordt het standaardpublicatieprofiel gebruikt.

### Geretourneerde waarde

Boolean

### Beschrijving

Methode, hiermee wordt een FLA-bestand gepubliceerd zonder het te openen. Met deze API wordt de FLA geopend in de headless-modus en wordt het SWF-bestand (of wat de instelling van het profiel ook is) gepubliceerd. De tweede parameter (publishProfile) is optioneel. De retourwaarde is een booleaanse waarde die aangeeft of het profiel al dan niet is gevonden. Wanneer de tweede parameter niet wordt opgegeven, wordt altijd de waarde true geretourneerd.

### Voorbeeld

In het volgende voorbeeld wordt de gebruiker gevraagd een FLA-bestand te selecteren en dit automatisch te publiceren met gebruik van het standaardpublicatieprofiel.

```
var uri = fl.browseForFileURL("select", "select a FLA file to publish");  
var publishProfileName = "Default";  
fl.publishDocument(uri, publishProfileName);
```

## fl.quit()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.quit( [bPromptIfNeeded] )
```

### Parameters

**bPromptIfNeeded** Een Booleaanse waarde die true is (standaard) als u wilt dat de gebruiker wordt gevraagd eventueel gewijzigde documenten op te slaan. Stel deze parameter in op false als u niet wilt dat de gebruiker wordt gevraagd gewijzigde documenten op te slaan. In het laatste geval zullen eventuele wijzigingen in geopende documenten worden genegeerd en wordt de toepassing direct afgesloten. Gebruik deze methode met uiterste voorzichtigheid, hoewel deze heel handig is voor batchverwerking. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt Flash afgesloten en wordt de gebruiker gevraagd eventueel gewijzigde documenten op te slaan.

### Voorbeeld

In het volgende voorbeeld wordt getoond hoe u kunt afsluiten met en zonder de vraag of gewijzigde documenten moeten worden opgeslagen:

```
// Quit with prompt to save any modified documents.  
fl.quit();  
fl.quit(true); // True is optional.  
// Quit without saving any files.  
fl.quit(false);
```

## fl.reloadEffects()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.reloadEffects();
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle effectdescriptors die in de map Configuration Effects van de gebruiker zijn gedefinieerd, opnieuw geladen. Hiermee kunt u de scripts tijdens het ontwikkelen snel wijzigen en hebt u een mechanisme om de effecten te verbeteren zonder dat u de toepassing opnieuw moet opstarten. Deze methode werkt het best als u deze in een opdracht gebruikt die zich in de map Commands bevindt.

### Voorbeeld

Het volgende voorbeeld is een eenregelig script dat u in de map Commands kunt plaatsen. Wanneer u effecten opnieuw wilt laden, gaat u naar het menu Opdrachten en voert u het script uit.

```
fl.reloadEffects();
```

## fl.reloadTools()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.reloadTools()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het deelvenster Gereedschappen opnieuw opgebouwd op basis van het bestand toolconfig.xml. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt. Gebruik deze methode wanneer u het deelvenster Gereedschappen opnieuw moet laden, bijvoorbeeld nadat u het JSFL-bestand hebt gewijzigd waarin een gereedschap wordt gedefinieerd dat al in het deelvenster aanwezig is.

### Voorbeeld

Het volgende voorbeeld is een eenregelig script dat u in de map Commands kunt plaatsen. Wanneer u het deelvenster Gereedschappen opnieuw moet laden, voert u het script vanuit het menu Opgdrachten uit.

```
fl.reloadTools();
```

## fl.removeEventListener()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.removeEventListener(eventType)
```

### Parameters

**eventType** Een tekenreeks die het type gebeurtenis aanduidt dat uit deze callback-functie moet worden verwijderd. Acceptabele waarden zijn "documentNew", "documentOpened", "documentClosed", "mouseMove", "documentChanged", "layerChanged" en "frameChanged".

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de gebeurtenislistener is verwijderd, `false` als de functie niet aan de lijst is toegevoegd met de methode `fl.addEventListener()`.

### Beschrijving

Hiermee wordt de registratie van een functie ongedaan gemaakt die met `fl.addEventListener()` was geregistreerd.



### Voorbeeld

In het volgende voorbeeld wordt de gebeurtenislistener verwijderd die is gekoppeld aan de gebeurtenis `documentClosed`:

```
fl.removeEventListener("documentClosed");
```

### Zie ook

[fl.addEventListener\(\)](#)

## fl.resetAS3PackagePaths()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.resetAS3PackagePaths()
```

### Parameters

Geen.

### Beschrijving

Methode; hiermee wordt de globale Classpath-instelling in het dialoogvenster ActionScript 3.0-instellingen teruggezet op de standaardwaarde. Als u de algemene ActionScript 2.0-Classpath opnieuw wilt instellen, gebruikt u

[fl.resetPackagePaths\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt de ActionScript 3,0 Classpath-instelling teruggezet op de standaardwaarde.

```
fl.resetAS3PackagePaths();
```

### Zie ook

[fl.as3PackagePaths](#)

## fl.resetPackagePaths()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.resetPackagePaths()
```

### Parameters

Geen.

### Beschrijving

Methode; hiermee wordt de globale Classpath-instelling in het dialoogvenster ActionScript 2.0-instellingen teruggezet op de standaardwaarde. Als u de algemene ActionScript 3.0-Classpath opnieuw wilt instellen, gebruikt u [fl.resetAS3PackagePaths\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt de ActionScript 2.0 Classpath-instelling teruggezet op de standaardwaarde.

```
fl.resetPackagePaths();
```

### Zie ook

[fl.packagePaths](#)

## fl.revertDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.revertDocument(documentObject)
```

### Parameters

**documentObject** Een [Object Document](#). Als *documentObject* naar het actieve document verwijst, wordt het venster Document mogelijk pas hersteld wanneer het script dat deze methode aanroept, volledig is uitgevoerd.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking Omkeren is voltooid, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven FLA-document teruggezet naar de laatst opgeslagen versie. In tegenstelling tot de menuoptie Bestand > Omkeren, wordt met deze methode geen waarschuwingsvenster weergegeven waarin de gebruiker wordt gevraagd de bewerking te bevestigen. Zie ook [document.revert\(\)](#) en [document.canRevert\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt het huidige FLA-document teruggezet naar de laatst opgeslagen versie; wijzigingen die sindsdien zijn aangebracht, gaan verloren.

```
fl.revertDocument(fl.getDocumentDOM());
```

## fl.runScript()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.runScript(fileURI [, funcName [, arg1, arg2, ...]])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de naam van het uit te voeren script aangeeft.

**funcName** Een tekenreeks die een functie identificeert die moet worden uitgevoerd in het JSFL-bestand dat is opgegeven in *fileURI*. Deze parameter is optioneel.

**arg** Een optionele parameter die een of meer argumenten opgeeft die moeten worden doorgegeven aan *funcname*.

### Geretourneerde waarde

Het resultaat van de functie als tekenreeks, als *funcName* is opgegeven, anders niets.

### Beschrijving

Methode; hiermee wordt een JavaScript-bestand uitgevoerd. Als een functie is opgegeven als een van de argumenten, wordt de functie uitgevoerd en ook eventuele code in het script dat niet binnen de functie is. De rest van de code in het script wordt uitgevoerd voordat de functie wordt uitgevoerd.

### Voorbeeld

Stel dat er in de hoofdmap op station C een scriptbestand is met de naam testScript.jsfl met de volgende inhoud:

```
function testFunc(num, minNum) {
    fl.trace("in testFunc: 1st arg: " + num + " 2nd arg: " + minNum);
}
for (i=0; i<2; i++) {
    fl.trace("in for loop i=" + i);
}
fl.trace("end of for loop");
// End of testScript.jsfl
```

Als u de volgende opdracht geeft,

```
fl.runScript("file:///C:/testScript.jsfl", "testFunc", 10, 1);
```

verschijnt deze informatie in het deelvenster Uitvoer:

```
in for loop i=0
in for loop i=1
end of for loop
in testFunc: 1st arg: 10 2nd arg: 1
```

U kunt ook testScript.jsfl aanroepen zonder een functie uit te voeren:

```
fl.runScript("file:///C:/testScript.jsfl");
```

Hiermee wordt het volgende weergegeven in het deelvenster Uitvoer:

```
in for loop i=0
in for loop i=1
end of for loop
```

## fl.saveAll()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.saveAll()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle geopende documenten opgeslagen.

Als een bestand nog niet eerder is opgeslagen of niet is gewijzigd sinds de laatste keer dat het is opgeslagen, wordt het bestand niet opgeslagen. Als u een niet-opgeslagen of ongewijzigd bestand wilt laten opslaan, gebruikt u

```
fl.saveDocumentAs()
```

### Voorbeeld

In het volgende voorbeeld worden alle geopende documenten opgeslagen die al eerder zijn opgeslagen en die zijn gewijzigd sinds de laatste keer dat ze zijn opgeslagen:

```
fl.saveAll();
```

### Zie ook

```
document.save(), document.saveAndCompact(), fl.saveDocument(), fl.saveDocumentAs()
```

## fl.saveDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.saveDocument(document [, fileURI])
```

### Parameters

**document** Een [Object Document](#) dat het document aangeeft dat moet worden opgeslagen. Als *document* `null` is, wordt het actieve document opgeslagen.

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de naam van het opgeslagen document aangeeft. Als de parameter *fileURI* `null` is of wordt weggelaten, wordt het document met de huidige naam opgeslagen. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het document is opgeslagen, anders `false`.

Als een bestand nog niet eerder is opgeslagen of niet is gewijzigd sinds de laatste keer dat het is opgeslagen, wordt het niet opgeslagen en wordt `false` geretourneerd. Als u een niet-opgeslagen of ongewijzigd bestand wilt laten opslaan, gebruikt u [fl.saveDocumentAs\(\)](#).

### Beschrijving

Methode; hiermee wordt het opgegeven document opgeslagen als FLA-document.

### Voorbeeld

In het volgende voorbeeld worden het huidige document en twee opgegeven documenten opgeslagen:

```
// Save the current document.  
alert(fl.saveDocument(fl.getDocumentDOM()));  
// Save the specified documents.  
alert(fl.saveDocument(fl.documents[0], "file:///C:/example1 fla");  
alert(fl.saveDocument(fl.documents[1], "file:///C:/example2 fla");
```

### Zie ook

[document.save\(\)](#), [document.saveAndCompact\(\)](#), [fl.saveAll\(\)](#), [fl.saveDocumentAs\(\)](#)

## fl.saveDocumentAs()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.saveDocumentAs(document)
```

### Parameters

**document** Een [Object Document](#) dat het document aangeeft dat moet worden opgeslagen. Als *document* `null` is, wordt het actieve document opgeslagen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking Opslaan als is uitgevoerd, anders `false`.

### Beschrijving

Methode; hiermee wordt het dialoogvenster Opslaan als weergegeven voor het opgegeven document.

### Voorbeeld

In het volgende voorbeeld wordt de gebruiker gevraagd het opgegeven document op te slaan, waarna een waarschuwingsbericht wordt weergegeven dat aangeeft of het document is opgeslagen:

```
alert(fl.saveDocumentAs(fl.documents[1]));
```

### Zie ook

[document.save\(\)](#), [document.saveAndCompact\(\)](#), [fl.saveAll\(\)](#), [fl.saveDocument\(\)](#)

## fl.scriptURI

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.scriptURI
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die staat voor het pad van het JSFL-script dat wordt uitgevoerd, uitgedrukt als bestand:/// URI. Als het script is aangeroepen vanuit `fl.runScript()`, staat deze eigenschap voor het pad van het direct bovenliggende script. Dat wil zeggen dat niet meerdere aanroepen van `fl.runScript()` worden doorlopen om het pad van het oorspronkelijke aanroepende script te vinden.

### Voorbeeld

In het volgende voorbeeld wordt het pad van het JSFL-script dat momenteel wordt uitgevoerd weergegeven in het deelvenster Uitvoer:

```
fl.trace(fl.scriptURI);
```

### Zie ook

[fl.runScript\(\)](#)

## fl.selectElement()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
fl.selectElement(elementObject, editMode)
```

### Parameters

**elementObject** Het [Object Element](#) dat u wilt selecteren.

**editMode** Een Booleaanse waarde die aangeeft of u het element wilt bewerken (`true`) of alleen wilt selecteren (`false`).

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het element is geselecteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt de selectie of bewerking van een element ingeschakeld. In het algemeen zult u deze methode gebruiken voor objecten die worden geretourneerd door `fl.findObjectInDocByName()` of `fl.findObjectInDocByType()`.

### Voorbeeld

In het volgende voorbeeld wordt een element geselecteerd met de naam "second text field" als dit wordt gevonden in het document:

**Object flash (fl)**

```

var nameToSearchFor = "second text field";
var doc = fl.getDocumentDOM();

// Start by viewing Scene 1 (index value of 0).
document.editScene(0);

// Search for element by name.
var results = fl.findObjectInDocByName(nameToSearchFor, doc);
if (results.length > 0) {
    // Select the first element found.
    // Pass false, so the symbolInstance you are searching for is selected.
    // If you pass true, the symbol instance will switch to edit mode.
    fl.selectElement(results[0], false);
    alert("success, found " + results.length + " objects")
}
else {
    alert("failed, no objects with name " + nameToSearchFor + " found");
}

```

**Zie ook**

[fl.findObjectInDocByName\(\)](#), [fl.findObjectInDocByType\(\)](#)

## fl.selectTool()

**Beschikbaarheid**

Flash CS3 Professional.

**Gebruik**

```
fl.selectTool(toolName)
```

**Parameters**

**toolName** Een tekenreeks waarmee de naam wordt opgegeven van het gereedschap dat moet worden geselecteerd. Zie “Beschrijving” hieronder voor informatie over geldige waarden voor deze parameter.

**Beschrijving**

Methode; hiermee wordt het opgegeven gereedschap geselecteerd in het deelvenster Gereedschappen. Acceptabele standaardwaarden voor *toolName* zijn "arrow", "bezierSelect", "freeXform", "fillXform", "lasso", "pen", "penplus", "penminus", "penmodify", "text", "line", "rect", "oval", "rectPrimitive", "ovalPrimitive", "polystar", "pencil", "brush", "inkBottle", "bucket", "eyeDropper", "eraser", "hand" en "magnifier".

Als u of een gebruiker aangepaste gereedschappen maakt, kunnen de namen van de gereedschappen ook worden doorgegeven als de parameter *toolName*. De lijst met gereedschapsnamen bevindt zich in het volgende bestand:

- Windows Vista:

```
opstartstation\Users\gebruikersnaam\Local Settings\Application Data\Adobe\Flash
CS3\taal\Configuration\Tools\toolConfig.xml
```

- Windows XP:

```
opstartstation\Documents and Settings\gebruikersnaam\Local Settings\Application Data\Adobe\Flash
CS3\taal\Configuration\Tools\toolConfig.xml
```

- Mac OS X:

Macintosh HD/Users/*gebruikersnaam*/Library/Application Support/Adobe/Flash  
CS3/*taal*/Configuration/Tools/toolConfig.xml

#### Voorbeeld

In het volgende voorbeeld wordt het gereedschap Pen geselecteerd.

```
fl.selectTool("pen");
```

#### Zie ook

[Object Tools](#), [Object ToolObj](#)

## fl.setActiveWindow()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
fl.setActiveWindow(document [, bActivateFrame])
```

#### Parameters

**document** Een [Object Document](#) dat het document aangeeft dat moet worden geselecteerd als het actieve venster.

**bActivateFrame** Een optionele parameter die door Flash en Fireworks wordt genegeerd en alleen aanwezig is omwille van de compatibiliteit met Dreamweaver.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt het actieve venster ingesteld op het opgegeven document. Deze methode wordt ook ondersteund door Dreamweaver en Fireworks. Als het document meerdere weergaven heeft (gemaakt door Venster > Venster dupliceren), wordt de laatst actieve weergave geselecteerd.

#### Voorbeeld

In het volgende voorbeeld worden twee manieren getoond om een specifiek document te activeren:

```
fl.setActiveWindow(fl.documents[0]);  
  
var theIndex = fl.findDocumentIndex("myFile fla");  
fl.setActiveWindow(fl.documents[theIndex]);
```

## fl.showIdleMessage()

#### Beschikbaarheid

Flash 8.



### Gebruik

```
fl.showIdleMessage(show)
```

### Parameters

**show** Een Booleaanse waarde die aangeeft of de waarschuwing voor een script dat te lang loopt, moet worden in- of uitgeschakeld.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee kunt u de waarschuwing voor een script dat te lang loopt, uitschakelen (geef `false` door voor *show*). U kunt dit bijvoorbeeld doen wanneer u batchbewerkingen verwerkt die lang duren. Als u de waarschuwing weer wilt inschakelen, geeft u de opdracht opnieuw en geeft u dit keer `true` door voor *show*.

### Voorbeeld

In het volgende voorbeeld ziet u hoe u de waarschuwing voor een script dat te lang loopt, kunt uitschakelen en weer inschakelen:

```
fl.showIdleMessage(false);  
var result = timeConsumingFunction();  
fl.showIdleMessage(true); ;  
var result = timeConsumingFunction();
```

## fl.sourcePath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fl.sourcePath
```

### Beschrijving

Eigenschap; een tekenreeks die een lijst met items bevat in het globale pad voor de bron van ActionScript 3.0. Hiermee wordt de locatie van ActionScript-klassenbestanden aangegeven. Items in de tekenreeks worden gescheiden door puntkomma's. In het ontwerp gereedschap worden de items opgegeven door Bewerken > Voorkeuren > ActionScript > ActionScript 3.0-instellingen te kiezen.

### Voorbeeld

In het volgende voorbeeld wordt de map `/Classes` toegevoegd aan het globale pad naar de bron van ActionScript 3.0:

```
fl.trace(fl.sourcePath);  
fl.sourcePath = "/Classes;" + fl.sourcePath;  
fl.trace(fl.sourcePath);
```

### Zie ook

[fl.flexSDKPath](#), [fl.externalLibraryPath](#), [fl.libraryPath](#), [document.sourcePath](#)

## fl.swfPanels

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fl.swfPanels
```

### Beschrijving

Alleen-lezen eigenschap; een array van geregistreerde objecten swfPanel (zie [Object swfPanel](#)). Een object swfPanel is geregistreerd als het ten minste eenmaal is geopend.

De positie van een deelvenster in de array staat voor de volgorde waarin het werd geopend. Als het eerste deelvenster dat is geopend TraceBitmap is en het tweede AnotherFunction, is `fl.swfPanels[0]` het swfPanel-object TraceBitmap, `fl.swfPanels[1]` het swfPanel-object AnotherFunction, enzovoort.

### Voorbeeld

Met de volgende code worden de naam en het pad van eventuele geregistreerde Window SWF-deelvensters in het deelvenster Uitvoer weergegeven:

```
if(fl.swfPanels.length > 0){
    for(x = 0; x < fl.swfPanels.length; x++){
        fl.trace("Panel: " + fl.swfPanels[x].name + " -- Path: " + fl.swfPanels[x].path);
    }
}
```

## fl.toggleBreakpoint()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
fl.toggleBreakPoint(String fileURI, int line, Boolean enable)
```

### Parameters

**fileURI** Een tekenreeks, de URI van het AS-bestand waarin het onderbrekingspunt moet worden in- of uitgeschakeld.

**line** Een geheel getal, het regelnummer waarbij het onderbrekingspunt moet worden in- of uitgeschakeld.

**enable** Booleaanse waarde, het onderbrekingspunt is ingeschakeld wanneer deze waarde is ingesteld op true. Wanneer de instelling false is, is het onderbrekingspunt uitgeschakeld.

### Beschrijving

Hiermee wordt op de desbetreffende regel een onderbrekingspunt voor het desbetreffende AS-bestand in- en uitgeschakeld. Als `enable` false is, wordt het momenteel op die regel opgeslagen onderbrekingspunt gewist.

### Voorbeeld

In het volgende voorbeeld wordt een onderbrekingspunt ingeschakeld bij regel 10 van het AS-bestand op C:\AS\breakpointTest.as:

```
fl.toggleBreakPoint("file:///C:/AS/breakpointTest.as", 10, 1);
```

## fl.tools

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.tools
```

### Beschrijving

Alleen-lezen eigenschap; een array van objecten Tools (zie [Object Tools](#)). Deze eigenschap wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

## fl.trace()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
fl.trace(message)
```

### Parameters

**message** Een tekenreeks die wordt weergegeven in het deelvenster Uitvoer.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een teksttekenreeks verzonden naar het deelvenster Uitvoer, afgesloten door een nieuwe regel. Het deelvenster Uitvoer verschijnt als dit nog niet zichtbaar is. Deze methode is identiek aan [outputPanel.trace\(\)](#) en werkt op dezelfde manier als de instructie `trace()` in ActionScript.

Als u een witregel wilt verzenden, gebruikt u `fl.trace("")` of `fl.trace("\n")`. U kunt de tweede opdracht inline gebruiken door `\n` op te nemen in de tekenreeks *message*.

### Voorbeeld

In het volgende voorbeeld worden meerdere tekstregels in het deelvenster Uitvoer weergegeven:

**Object flash (fl)**

```
fl.outputPanel.clear();
fl.trace("Hello World!!!");
var myPet = "cat";
fl.trace("\nI have a " + myPet);
fl.trace("");
fl.trace("I love my " + myPet);
fl.trace("Do you have a " + myPet +"?");
```

## fl.version

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
fl.version
```

**Beschrijving**

Alleen-lezen eigenschap; de lange-tekenreeksversie van het ontwerpgereedschap van Flash, inclusief platform.

**Voorbeeld**

In het volgende voorbeeld wordt de versie van het ontwerpgereedschap van Flash weergegeven in het deelvenster Uitvoer:

```
alert(fl.version); // For example, WIN 10,0,0,540
```

## fl.xmlui

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
fl.xmlui
```

**Beschrijving**

Alleen-lezen eigenschap; een [Object XMLUI](#). Met deze eigenschap kunt u XMLUI-eigenschappen in een dialoogvenster XMLUI ophalen en instellen en kunt u het dialoogvenster programmatisch accepteren of annuleren.

**Voorbeeld**

Zie [Object XMLUI](#).

# Hoofdstuk 18: Object FLfile

## Beschikbaarheid

Flash MX 2004 7.2.

## Beschrijving

Met het object FLfile kunt u Flash-uitbreidingen schrijven die bestanden en mappen op het lokale systeem kunnen openen, wijzigen en verwijderen. De FLfile-API heeft de vorm van een uitbreiding van de JavaScript-API. Deze uitbreiding wordt een *gedeelde bibliotheek* genoemd en bevindt zich in de volgende map:

- Windows Vista:

```
opstartstation \Gebruikers\gebruikersnaam \Local Settings\Application Data\Adobe\Flash CS3\taal
\Configuration\External Libraries\FLfile.dll
```

- Windows XP:

```
opstartstation \Documents and Settings\gebruikersnaam \Local Settings\Application Data\Adobe\Flash CS3\taal
\Configuration\External Libraries\FLfile.dll
```

- Mac OS X:

```
Macintosh HD/Users/gebruikersnaam /Library/Application Support/Adobe/Flash
CS3/taal/Configuration/External Libraries
```

**Opmerking:** De gedeelde JavaScript-API-bibliotheken moeten niet worden verward met de gedeelde bibliotheken die symbolen in uw Flash-documenten bevatten. Het zijn twee heel verschillende dingen.

De FLfile-methoden werken met bestanden of mappen op de schijf. Elke methode bevat daarom een of meer parameters om de locatie van een map of bestand op te geven. De locatie van het bestand of de map wordt uitgedrukt als tekenreeks die sterk lijkt op een URL van een website. Dit heet een bestands-URI (Uniform Resource Identifier) en is volgt opgebouwd (inclusief de aanhalingstekens):

```
"file:///drive|/folder 1/folder 2/.../filename"
```

Als u bijvoorbeeld een map met de naam config op de C-schijf wilt maken en in de map Program Files/MyApp wilt plaatsen, gebruikt u de volgende opdracht:

```
FLfile.createFolder("file:///C:/Program Files/MyApp/config");
```

Als u daarna een bestand met de naam config.ini in die map wilt plaatsen, gebruikt u de volgende opdracht:

```
FLfile.write("file:///C:/Program Files/MyApp/config/config.ini", "");
```

Als u een map op de Macintosh wilt maken, kunt u de volgende opdracht gebruiken:

```
FLfile.createFolder("file:///Macintosh/MyApp/config");
```

## Overzicht van methoden

Voor het object FLfile kunnen de volgende methoden worden gebruikt:

Method	Beschrijving
<code>FLfile.copy()</code>	Hiermee wordt een bestand gekopieerd.
<code>FLfile.createFolder()</code>	Hiermee worden een of meer mappen gemaakt.
<code>FLfile.exists()</code>	Hiermee wordt vastgesteld of een bestand of map bestaat.
<code>FLfile.getAttributes()</code>	Hiermee wordt nagegaan of een bestand schrijfbaar, alleen-lezen, verborgen, zichtbaar of een systeemmap is.
<code>FLfile.getCreationDate()</code>	Hiermee wordt aangegeven hoeveel seconden zijn verstreken sinds 1 januari 1970 tot aan het tijdstip waarop het bestand of de map is gemaakt.
<code>FLfile.getCreationDateObj()</code>	Hiermee wordt de datum opgehaald waarop een bestand of map is gemaakt.
<code>FLfile.getModificationDate()</code>	Hiermee wordt aangegeven hoeveel seconden zijn verstreken sinds 1 januari 1970 tot aan het tijdstip waarop het bestand of de map voor het laatst is gewijzigd.
<code>FLfile.getModificationDateObj()</code>	Hiermee wordt de datum opgehaald waarop een bestand of map voor het laatst is gewijzigd.
<code>FLfile.getSize()</code>	Hiermee wordt de grootte van een bestand opgehaald.
<code>FLfile.listFolder()</code>	Hiermee wordt de inhoud van een map weergegeven.
<code>FLfile.platformPathToURI()</code>	Hiermee wordt een bestandsnaam van een platformspecifieke notatie omgezet in bestand:/// URI.
<code>FLfile.read()</code>	Hiermee wordt de inhoud van een bestand gelezen.
<code>FLfile.remove()</code>	Hiermee wordt een bestand of map verwijderd.
<code>FLfile.setAttributes()</code>	Hiermee wordt een bestand of map alleen-lezen, overschrijfbaar, verborgen of zichtbaar gemaakt.
<code>FLfile.uriToPlatformPath()</code>	Hiermee wordt een bestandsnaam uitgedrukt als bestand:/// URI omgezet in een platformspecifieke notatie.
<code>FLfile.write()</code>	Hiermee wordt een bestand gemaakt, wordt naar een bestand geschreven of wordt iets aan het bestand toegevoegd.

## FLfile.copy()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.copy(fileURI, copyURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand dat u wilt kopiëren aangeeft.

**copyURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de locatie en naam van het gekopieerde bestand aangeeft.

### Geretournerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol is, anders `false`.

### Beschrijving

Methode; hiermee wordt een bestand van één locatie naar een andere gekopieerd. Deze methode retourneert `false` als `copyURI` al bestaat.

### Voorbeeld

In het volgende voorbeeld wordt een reservekopie van een configuratiebestand `config.ini` gemaakt, en wordt dit in dezelfde map onder een nieuwe naam opgeslagen:

```
var originalFileURI="file:///C:/Program Files/MyApp/config.ini";
var newFileURI="file:///C:/Program Files/MyApp/config_backup.ini";
FLfile.copy(originalFileURI, newFileURI);
```

U kunt dezelfde taak ook met één enkele opdracht uitvoeren:

```
FLfile.copy("file:///C|:/Program Files/MyApp/config.ini", file:///C|/Program
Files/MyApp/config_backup.ini");
```

## FLfile.createFolder()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.createFolder(folderURI)
```

### Parameters

`folderURI` Een map-URI die de mapstructuur aangeeft die u wilt maken.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol is; `false` als `folderURI` al bestaat.

### Beschrijving

Methode; hiermee worden één of meerdere mappen op de opgegeven locatie gemaakt.

U kunt meerdere mappen tegelijk aanmaken. Met de volgende opdracht maakt u bijvoorbeeld tegelijk de map `MyData` en `TempData`, als deze nog niet bestonden:

```
FLfile.createFolder("file:///c|/MyData/TempData")
```

### Voorbeeld

In het volgende voorbeeld worden een map en een submap in de configuratiemap (`fl.configURI`) gemaakt:

```
fl.trace(FLfile.createFolder(fl.configURI+"folder01/subfolder01"));
```

In het volgende voorbeeld wordt geprobeerd een map met de naam `tempFolder` in de hoofdmap op de C-drive te maken. Een waarschuwingsvenster geeft aan of dit is gelukt:

```
var folderURI = "file:///c:/tempFolder";
if (FLfile.createFolder(folderURI)) {
    alert("Created " + folderURI);
}
else {
    alert(folderURI + " already exists");
}
```

#### Zie ook

[FLfile.remove\(\)](#), [FLfile.write\(\)](#)

## FLfile.exists()

#### Beschikbaarheid

Flash MX 2004 7.2.

#### Gebruik

`FLfile.exists(fileURI)`

#### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand dat u wilt controleren aangeeft.

#### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol is, anders `false`.

#### Beschrijving

Methode; hiermee wordt aangegeven of een opgegeven bestand bestaat. Als u een map en bestandsnaam opgeeft, moet de map al bestaan. Zie [FLfile.createFolder\(\)](#) voor informatie over het maken van mappen.

#### Voorbeelden

In het volgende voorbeeld wordt in de map temp naar het bestand mydata.txt gezocht. Een waarschuwingsvenster geeft aan of het bestand bestaat:

```
var fileURI = "file:///c:/temp/mydata.txt";
if (FLfile.exists(fileURI)) {
    alert( fileURI + " exists.");
}
else {
    alert( fileURI + " does not exist.");
}
```

In het volgende voorbeeld wordt gecontroleerd of een vereist configuratiebestand in de map MyApplication staat. Als het bestand niet bestaat, wordt het gemaakt.

```
var configFile = "file:///C:/MyApplication/config.ini";
if (!FLfile.exists(configFile)) {
    FLfile.write(configFile, "");
}
}
```



### Zie ook

[FLfile.write\(\)](#)

## FLfile.getAttributes()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.getAttributes(fileOrFolderURI)
```

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand of de map aangeeft waarvan u de kenmerken wilt ophalen.

### Geretourneerde waarde

Een tekenreeks die de kenmerken van het opgegeven bestand of de map vertegenwoordigt.

Wanneer het bestand of de map niet bestaat, is het resultaat niet te voorspellen. Het wordt aangeraden [FLfile.exists\(\)](#) aan te roepen voordat u deze methode gebruikt.

### Beschrijving

Methode; hiermee wordt een tekenreeks geretourneerd die de kenmerken van het opgegeven bestand of de map vertegenwoordigt, of een lege tekenreeks wanneer het bestand geen specifieke kenmerken bevat (het is dus geen alleen-lezen, niet verborgen, enzovoort). Controleer met [FLfile.exists\(\)](#) of een bestand of map bestaat voordat u deze methode gebruikt.

Tekens in de tekenreeks vertegenwoordigen de kenmerken als volgt:

- **R** — *fileOrFolderURI* is alleen-lezen
- **D** — *fileOrFolderURI* is een map (directory)
- **H** — *fileOrFolderURI* is verborgen (alleen Windows)
- **S** — *fileOrFolderURI* is een systeembestand of -map (alleen Windows)
- **A** — *fileOrFolderURI* is gereed voor archivering (alleen Windows)

Als *fileOrFolderURI* bijvoorbeeld een verborgen map is, wordt de tekenreeks "DH" geretourneerd.

### Voorbeeld

In het volgende voorbeeld worden de kenmerken van het bestand mydata.txt opgehaald. Er verschijnt een waarschuwingsvenster als het bestand alleen-lezen is.

```
var URI = "file:///c:/temp/mydata.txt";  
if (FLfile.exists(URI)) {  
    var attr = FLfile.getAttributes(URI);  
    if (attr && (attr.indexOf("R") != -1)) { // Returned string contains R.  
        alert(URI + " is read only!");  
    }  
}
```

### Zie ook

[FLfile.setAttributes\(\)](#)

## FLfile.getCreationDate()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.getCreationDate(fileOrFolderURI)
```

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand of de map aangeeft waarvan u de aanmaakdatum en het aanmaaktijdstip als hexadecimale tekenreeks wilt ophalen.

### Geretourneerde waarde

Een tekenreeks met een hexadecimaal getal voor het aantal seconden dat is verstreken sinds 1 januari 1970 tot aan het tijdstip waarop het bestand of de map is gemaakt, of "00000000" als het bestand of de map niet bestaat.

### Beschrijving

Methode; geeft aan hoeveel seconden zijn verstreken sinds 1 januari 1970 tot aan het tijdstip waarop het bestand of de map is gemaakt. Deze methode wordt vooral gebruikt de datum van aanmaak of wijziging van bestanden of mappen te vergelijken.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of een bestand sinds het is gemaakt nog werd gewijzigd:

```
// Make sure the specified file exists
var fileURI = "file:///C:/MyApplication/MyApp fla";
var creationTime = FLfile.getCreationDate(fileURI);
var modificationTime = FLfile.getModificationDate(fileURI);
if ( modificationTime > creationTime ) {
    alert("The file has been modified since it was created.");
}
else {
    alert("The file has not been modified since it was created.");
}
```

### Zie ook

[FLfile.getCreationDateObj\(\)](#), [FLfile.getModificationDate\(\)](#)

## FLfile.getCreationDateObj()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.getCreationDateObj(fileOrFolderURI)
```

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand of de map aangeeft waarvan u de aanmaakdatum en het aanmaaktijdstip als JavaScript Date-object wilt ophalen.

### Geretourneerde waarde

Een JavaScript Date-object dat de datum en het tijdstip vertegenwoordigt waarop het opgegeven bestand of de map is gemaakt. Als het bestand niet bestaat, bevat het object informatie die aangeeft dat het bestand of de map op 31 december 1969, om middernacht GMT is gemaakt.

### Beschrijving

Methode; hiermee wordt een JavaScript Date-object geretourneerd dat de datum en het tijdstip vertegenwoordigt waarop het opgegeven bestand of de map is gemaakt.

### Voorbeeld

In het volgende voorbeeld wordt de datum waarop een bestand is gemaakt leesbaar weergegeven in het deelvenster Uitvoer :

```
// Make sure the specified file exists.  
var file1Date = FLfile.getCreationDateObj("file:///c:/temp/file1.txt");  
fl.trace(file1Date);
```

### Zie ook

[FLfile.getCreationDate\(\)](#), [FLfile.getModificationDateObj\(\)](#)

## FLfile.getModificationDate()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.getModificationDate(fileOrFolderURI)
```

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand aangeeft waarvan u de datum en het tijdstip van wijziging als hexadecimale tekenreeks wilt ophalen.

### Geretourneerde waarde

Een tekenreeks met een hexadecimaal getal voor het aantal seconden dat is verstreken sinds 1 januari 1970 tot aan het tijdstip waarop het bestand of de map voor het laatst werd gewijzigd, of "00000000" als het bestand of de map niet bestaat.

### Beschrijving

Methode; geeft aan hoeveel seconden zijn verstreken sinds 1 januari 1970 tot aan het tijdstip waarop het bestand of de map voor het laatst is gewijzigd. Deze methode wordt vooral gebruikt de datum van aanmaak of wijziging van bestanden of mappen te vergelijken.

### Voorbeeld

In het volgende voorbeeld worden de datums van wijziging van twee bestanden vergeleken en wordt bepaald welk van de twee bestanden het laatst is gewijzigd:

```
// Make sure the specified files exist.
file1 = "file:///C:/MyApplication/MyApp fla";
file2 = "file:///C:/MyApplication/MyApp.as";
modificationTime1 = FLfile.getModificationDate(file1);
modificationTime2 = FLfile.getModificationDate(file2) ;
if(modificationTime1 > modificationTime2) {
    alert("File 2 is older than File 1") ;
}
else if(modificationTime1 < modificationTime2) {
    alert("File 1 is older than File 2") ;
}
else {
    alert("File 1 and File 2 were saved at the same time") ;
}
```

### Zie ook

[FLfile.getCreationDate\(\)](#), [FLfile.getModificationDateObj\(\)](#)

## FLfile.getModificationDateObj()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

`FLfile.getModificationDateObj(fileOrFolderURI)`

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand of de map aangeeft waarvan u de datum en het tijdstip van wijziging als JavaScript Date-object wilt ophalen.

### Geretoureerde waarde

Een JavaScript Date-object dat de datum en het tijdstip vertegenwoordigt waarop het opgegeven bestand of de map voor het laatst werd gewijzigd. Als het bestand niet bestaat, bevat het object informatie die aangeeft dat het bestand of de map op 31 december 1969, om middernacht GMT is gemaakt.

### Beschrijving

Methode; hiermee wordt een JavaScript Date-object geretourneerd dat de datum en het tijdstip vertegenwoordigt waarop het opgegeven bestand of de map voor het laatst werd gewijzigd.

### Voorbeeld

In het volgende voorbeeld wordt de datum waarop een bestand voor het laatst is gewijzigd leesbaar weergegeven in het deelvenster Uitvoer:

```
// Make sure the specified file exists.  
var file1Date = FLfile.getModificationDateObj("file:///c:/temp/file1.txt");  
trace(file1Date);
```

### Zie ook

[FLfile.getCreationDateObj\(\)](#), [FLfile.getModificationDate\(\)](#)

## FLfile.getSize()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.getSize(fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand aangeeft waarvan u de grootte wilt ophalen.

### Geretourneerde waarde

Een geheel getal dat de grootte van het opgegeven bestand in bytes vertegenwoordigt, of 0 als het bestand niet bestaat.

### Beschrijving

Methode; hiermee wordt een geheel getal geretourneerd dat de grootte van het opgegeven bestand in bytes vertegenwoordigt, of 0 als het bestand niet bestaat. Als de retourwaarde 0 is, kunt u met [FLfile.exists\(\)](#) bepalen of dit een bestand met een lengte van nul bytes is of dat het bestand niet bestaat.

Deze methode retourneert alleen correcte grootten voor bestanden met een grootte die kleiner is dan of gelijk is aan 2GB.

### Voorbeeld

In het volgende voorbeeld wordt de grootte van het bestand mydata.txt opgeslagen in de variabele fileSize:

```
var URL = "file:///c:/temp/mydata.txt";  
var fileSize = FLfile.getSize(URL);
```

## FLfile.listFiles()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.listFiles(folderURI [, filesOrDirectories])
```

### Parameters

**folderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de map aangeeft waarvan u de inhoud wilt ophalen. U kunt jokertekens opnemen in *folderURI*. Geldige jokertekens zijn \* (voor één of meer tekens) en ? (voor één teken).

**filesOrDirectories** Een optionele tekenreeks die aangeeft of alleen bestandsnamen of alleen mapnamen (directory's) moeten worden geretourneerd. Als u geen waarde opgeeft, worden zowel bestands- als mapnamen geretourneerd. Acceptabele waarden zijn "files" en "directories".

### Geretourneerde waarde

Een array met tekenreeksen die de inhoud van de map vertegenwoordigen. Er wordt een lege array geretourneerd als de map niet bestaat of als er geen bestanden of mappen aan de opgegeven criteria voldoen.

### Beschrijving

Methode; hiermee wordt een array met tekenreeksen geretourneerd die de inhoud van de map vertegenwoordigen.

### Voorbeelden

Het volgende voorbeeld retourneert drie arrays. De eerste vertegenwoordigt alle bestanden in de map C:\temp, de tweede alle mappen in de map C:\temp en de derde de bestanden en mappen in de map C:\temp:

```
var fileURI = "file:///C:/temp/" ;
var folderURI = "file:///C:/temp" ;
var fileList1 = FLfile.listFolder(fileURI, "files"); // files
var fileList2 = FLfile.listFolder(folderURI, "directories"); //folders
var fileList3 = FLfile.listFolder(folderURI); //files and folders
fl.trace("Files: " + fileList1);
fl.trace("");
fl.trace("Folders: " + fileList2);
fl.trace("");
fl.trace("Files and folders: " + fileList3);
```

In het volgende voorbeeld wordt een array geretourneerd van alle tekstbestanden (.txt) in de map temp en wordt de lijst weergegeven in een waarschuwingsvenster:

```
var folderURI = "file:///c:/temp";
var fileMask = "*.txt";
var list = FLfile.listFolder(folderURI + "/" + fileMask, "files");
if (list) {
    alert(folderURI + " contains: " + list.join(" "));
}
```

In het volgende voorbeeld wordt een bestandsmasker gebruikt in de opgegeven *folderURI* om de namen van alle uitvoerbare bestanden in de Windows-toepassingsmap te retourneren:

```
var executables = FLfile.listFolder("file:///C:/WINDOWS/*.exe", "files");
alert(executables.join("\n"));
```

## FLfile.platformPathToURI()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
FLfile.platformPathToURI(fileName)
```

### Parameters

**fileName** Een tekenreeks, uitgedrukt in een platformspecifieke notatie, die de bestandsnaam aangeeft die u wilt converteren.

### Geretoureerde waarde

Een tekenreeks uitgedrukt als bestand:/// URI.

### Beschrijving

Methode; hiermee wordt een bestandsnaam van een platformspecifieke notatie omgezet in bestand:/// URI.

### Voorbeeld

In het volgende voorbeeld wordt een bestandsnaam van een platformspecifieke notatie omgezet in bestand:/// URI. Deze wordt vervolgens doorgegeven aan `outputPanel.save()`:

```
var myFilename = "C:\\outputPanel.txt";  
var myURI=FLfile.platformPathToURI(myFilename);  
fl.outputPanel.save(myURI);
```

### Zie ook

[FLfile.uriToPlatformPath\(\)](#)

## FLfile.read()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.read()
```

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand of de map aangeeft waarvan u de kenmerken wilt ophalen.

### Geretoureerde waarde

De inhoud van het opgegeven bestand als tekenreeks, of `null` als het lezen mislukt.

### Beschrijving

Methode; hiermee wordt de inhoud geretourneerd van het opgegeven bestand als tekenreeks, of `null` als het lezen mislukt.

### Voorbeelden

In het volgende voorbeeld wordt het bestand `mydata.txt` gelezen en wordt, als dit succesvol was, een waarschuwingsvenster weergegeven met de inhoud van het bestand.

```
var fileURI = "file:///c:/temp/mydata.txt";
var str = FLfile.read( fileURI);
if (str) {
    alert( fileURL + " contains: " + str);
}
```

In het volgende voorbeeld wordt de ActionScript-code van een klassebestand gelezen en opgeslagen in de variabele code:

```
var classFileURI = "file:///C:/MyApplication/TextCarousel.as";
var code = FLfile.read(classFileURI);
```

## FLfile.remove()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.remove(fileOrFolderURI)
```

### Parameters

**fileOrFolderURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand of de map aangeeft dat of die u wilt wissen (verwijderen).

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol is, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven bestand of de opgegeven map verwijderd. Wanneer er in de map bestanden staan, worden deze ook verwijderd. Bestanden met het kenmerk R (alleen-lezen) kunnen niet worden verwijderd.

### Voorbeelden

In het volgende voorbeeld wordt een gebruiker gewaarschuwd dat een bestand bestaat. Indien de gebruiker dit wil, wordt het verwijderd:

```
var fileURI = prompt ("Enter file/folder to be deleted: ", "file:///c:/temp/delete.txt");
if (FLfile.exists(fileURI)) {
    var confirm = prompt("File exists. Delete it? (y/n)", "y");
    if (confirm == "y" || confirm == "Y") {
        if(FLfile.remove(fileURI)) {
            alert(fileURI + " is deleted.");
        }
        else {
            alert("fail to delete " + fileURI);
        }
    }
}
else {
    alert(fileURI + " does not exist");
}
```

In het volgende voorbeeld wordt een door een toepassing gemaakt configuratiebestand verwijderd:



```
if(FLfile.remove("file:///C:/MyApplication/config.ini")) {
    alert("Configuration file deleted");
}
```

In het volgende voorbeeld worden de map Configuration en de inhoud ervan verwijderd:

```
FLfile.remove("file:///C:/MyApplication/Configuration/");
```

### Zie ook

[FLfile.createFolder\(\)](#), [FLfile.getAttributes\(\)](#)

## FLfile.setAttributes()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.setAttributes(fileURI, strAttrs)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand aangeeft waarvan u de kenmerken wilt instellen.

**strAttrs** Een tekenreeks die de waarden aangeeft voor een of meer kenmerken die u wilt instellen. Zie de sectie "Beschrijving" hieronder voor geldige waarden voor *strAttrs*.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol is.

**Opmerking:** Wanneer het bestand of de map niet bestaat, is het resultaat niet te voorspellen. Het wordt aangeraden [FLfile.exists\(\)](#) aan te roepen voordat u deze methode gebruikt.

### Beschrijving

Methode; hiermee worden voor het opgegeven bestand kenmerken op systeemniveau aangegeven.

De volgende waarden zijn geldig voor *strAttrs*:

- `N` — Geen specifieke kenmerken (geen alleen-lezen, niet verborgen, enzovoort)
- `A` — Gereed voor archivering (alleen Windows)
- `R` — Alleen-lezen (op een Macintosh betekent alleen-lezen "vergrendeld")
- `W` — Overschrijfbaar (heeft hogere prioriteit dan `R`)
- `H` — Verborgen (alleen Windows)
- `V` — Zichtbaar (heeft hogere prioriteit dan `H`, alleen Windows)

Als u zowel `R` als `W` opneemt in *strAttrs*, wordt `R` genegeerd en is het bestand overschrijfbaar. Op dezelfde wijze wordt als u `H` en `V` doorgeeft, `H` genegeerd en is het bestand zichtbaar.

Wanneer u er zeker van wilt zijn dat het archiefkenmerk niet staat ingesteld, gebruik dan vóór het instellen van kenmerken deze opdracht met de parameter `N`. Er mag dan geen directe tegenhanger voor `A` zijn die het archiefkenmerk uitschakelt.

### Voorbeelden

In het volgende voorbeeld wordt ingesteld dat het bestand mydata.txt alleen-lezen en verborgen is. Dit heeft geen gevolgen voor het archiefkenmerk.

```
var URI = "file:///c:/temp/mydata.txt";
if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "RH");
}
```

In het volgende voorbeeld wordt ingesteld dat het bestand mydata.txt alleen-lezen en verborgen is. Hierdoor wordt gegarandeerd dat het archiefkenmerk niet is ingesteld.

```
var URI = "file:///c:/temp/mydata.txt";

if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "N");
    FLfile.setAttributes(URI, "RH");
}
```

### Zie ook

[FLfile.getAttributes\(\)](#)

## FLfile.uriToPlatformPath()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
FLfile.uriToPlatformPath(fileURI)
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die de bestandsnaam aangeeft die u wilt converteren.

### Geretourneerde waarde

Een tekenreeks die een platformspecifiek pad vertegenwoordigt.

### Beschrijving

Methode; hiermee wordt een bestandsnaam die wordt uitgedrukt als bestand:/// URI omgezet in een platformspecifieke notatie.

### Voorbeeld

In het volgende voorbeeld wordt bestand:/// URI omgezet in een platformspecifieke notatie:

```
var dir =(fl.configDirectory);
var URI = FLfile.platformPathToURI(dir);
fl.trace(URI == fl.configURI); // displays "true"
```

### Zie ook

[FLfile.platformPathToURI\(\)](#)

## FLfile.write()

### Beschikbaarheid

Flash MX 2004 7.2.

### Gebruik

```
FLfile.write(fileURI, textToWrite, [ , strAppendMode])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het bestand aangeeft waarin u wilt schrijven.

**textToWrite** Een tekenreeks die de tekst vertegenwoordigt die u in het bestand wilt plaatsen.

**strAppendMode** Een optionele tekenreeks met de waarde "append", die aangeeft dat u *textToWrite* aan het bestaande bestand wilt toevoegen. Als u geen waarde opgeeft, wordt *fileURI* overschreven door *textToWrite*.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de bewerking succesvol is, anders `false`.

### Beschrijving

Methode; hiermee wordt de opgegeven tekenreeks naar het opgegeven bestand geschreven (als UTF-8). Als het opgegeven bestand niet bestaat, wordt het gemaakt. De map waarin u het bestand plaatst, moet bestaan voordat u deze methode gebruikt. Gebruik `FLfile.createFolder()` om mappen te maken.

### Voorbeeld

In het volgende voorbeeld wordt geprobeerd de tekenreeks "xxx" in het bestand mydata.txt te schrijven. Als dit lukt, wordt een waarschuwingsbericht weergegeven. Vervolgens wordt geprobeerd de tekenreeks "aaa" aan het bestand toe te voegen. Als dit lukt, wordt een tweede waarschuwingsbericht weergegeven. Nadat dit script is uitgevoerd, bevat het bestand mydata.txt alleen de tekst "xxxaaa".

```
var URI = "file:///c:/temp/mydata.txt";
if (FLfile.write(URI, "xxx")) {
    alert("Wrote xxx to " + URI);
}
if (FLfile.write(URI, "aaa", "append")) {
    alert("Appended aaa to " + fileURI);
}
```

### Zie ook

`FLfile.createFolder()`, `FLfile.exists()`

# Hoofdstuk 19: Object folderItem

**Overerving** [Object Item](#) > Object folderItem

**Beschikbaarheid**

Flash MX 2004.

**Beschrijving**

Het object folderItem is een subklasse van het object Item. Er zijn geen unieke methoden of eigenschappen voor folderItem. Zie [Object Item](#).

# Hoofdstuk 20: Object fontItem

**Overerving** [Object Item](#) > object fontItem

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object fontItem is een subklasse van het object Item (zie [Object Item](#)).

## Overzicht van eigenschappen

Naast de eigenschappen van het object Item zijn de volgende eigenschappen beschikbaar voor het object fontItem:

Eigenschap	Beschrijving
<a href="#">fontItem.bitmap</a>	Hiermee wordt opgegeven of het item Font wordt omgezet naar bitmap.
<a href="#">fontItem.bold</a>	Hiermee wordt opgegeven of het item Font vetgedrukt is.
<a href="#">fontItem.embeddedCharacters</a>	Hiermee worden de in te sluiten gegevens opgegeven.
<a href="#">fontItem.embedRanges</a>	Hiermee wordt opgegeven welke items kunnen worden geselecteerd in het dialoogvenster Lettertype insluiten.
<a href="#">fontItem.embedVariantGlyphs</a>	Hiermee wordt opgegeven of bij publicatie van een SWF-bestand variantglyphs in het lettertype moeten worden gebruikt.
<a href="#">fontItem.font</a>	De naam van het apparaatlettertype dat aan het item Font is gekoppeld.
<a href="#">fontItem.isDefineFont4Symbol</a>	Hiermee wordt de opmaak van het lettertype opgegeven die bij publicatie van een SWF-bestand wordt uitgevoerd.
<a href="#">fontItem.italic</a>	Hiermee wordt opgegeven of het item Font cursief is.
<a href="#">fontItem.size</a>	De grootte van het item Font in punten.

## fontItem.bitmap

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`fontItem.bitmap`

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of het lettertype-item wordt omgezet naar bitmap (`true`) of niet (`false`).

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, wordt met de volgende code `true` weergegeven in het deelvenster Uitvoer als het item is omgezet naar bitmap, `false` als dit niet zo is:

**Object fontItem**

```
var theItem = fl.getDocumentDOM().library.items[0];  
fl.trace("bitmap: "+ theItem.bitmap);
```

## fontItem.bold

**Beschikbaarheid**

Flash CS4 Professional.

**Gebruik**

```
fontItem.bold
```

**Beschrijving**

Eigenschap; een Booleaanse waarde die opgeeft of het lettertype-item vet is (`true`) of niet (`false`).

**Voorbeeld**

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, wordt met de volgende code `true` weergegeven in het deelvenster Uitvoer als het item vet is, en wordt `false` als dit niet zo is en wordt het lettertype-item vervolgens ingesteld op vet:

```
var theItem = fl.getDocumentDOM().library.items[0];  
fl.outputPanel.clear();  
fl.trace("bold: "+ theItem.bold);  
theItem.bold=true;  
fl.trace("bold: "+ theItem.bold);
```

## fontItem.embeddedCharacters

**Beschikbaarheid**

Flash CS5 Professional.

**Gebruik**

```
fontItem.embeddedCharacters
```

**Beschrijving**

Eigenschap; een tekenreekswaarde waarmee u tekens kunt opgeven die in een SWF-bestand moeten worden ingesloten, zodat de tekens niet aanwezig hoeven te zijn op de apparaten waarop het SWF-bestand uiteindelijk wordt afgespeeld. Deze eigenschap biedt dezelfde functionaliteit als het dialoogvenster Lettertype insluiten.

Deze eigenschap kan ook worden gelezen, zodat u kunt zien welke tekens er voor een bepaald lettertype-item zijn opgegeven in het dialoogvenster Lettertype insluiten.

**Voorbeeld**

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, worden met volgende code de tekens a, b en c ingesloten.

```
fl.getDocumentDOM().library.items[0].embeddedCharacters = "abc";
```

## fontItem.embedRanges

### Beschikbaarheid

Flash CS5 Professional.

### Gebruik

`fontItem.embedRanges`

### Beschrijving

Eigenschap; een tekenreekswaarde die een reeks opgeeft met gescheiden gehele getallen, die overeenkomen met items die kunnen worden geselecteerd in het dialoogvenster Lettertype insluiten.

Deze eigenschap kan ook worden gelezen, zodat u kunt zien welke tekens er voor een bepaald lettertype-item zijn opgegeven in het dialoogvenster Lettertype insluiten.

**Opmerking:** De bereiknummers komen overeen met de nummers in het bestand `FontEmbedding/UnicodeTables.xml` in de configuratiemap.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, worden met de volgende code de bereiken ingesloten die worden aangeduid door de gehele getallen 1, 3 en 7.

```
fl.getDocumentDOM().library.items[0].embedRanges = "1|3|7";
```

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, worden met de volgende code de bereiken die moeten worden ingesloten, opnieuw ingesteld.

```
fl.getDocumentDOM().library.items[0].embedRanges = "";
```

## fontItem.embedVariantGlyphs

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`fontItem.embedVariantGlyphs`

### Beschrijving

**Opmerking:** Hoewel deze eigenschap beschikbaar is in Flash CS 5 Professional, heeft deze geen effect wanneer hij wordt toegepast op TLF-tekst (Text Layout Framework). Vanaf Flash Professional CS5 worden variantglyphs altijd ingesloten in lettertypen die bij TLF-tekst worden gebruikt. De `flash.text.engine` (FTE) waarnaar hier wordt verwezen, is alleen beschikbaar in Flash Professional CS4.

Eigenschap; een Booleaanse waarde die aangeeft of bij publicatie van een SWF-bestand variantglyphs moeten worden gebruikt in het lettertype (`true`) of niet (`false`). Als u deze waarde instelt op `true`, wordt het SWF-bestand groter. De standaardwaarde is `false`.

In sommige talen worden tekenglyphs tijdens het typen dynamisch vervangen, bijvoorbeeld in het Thais, Arabisch, Hebreeuws en Grieks. Als u tekst in een van deze talen opmaakt of invoert, stelt u deze eigenschap in op `true`.

### Voorbeelden

Lettertypesymbolen die geschikt zijn voor flash.text-API's staan in de Bibliotheek, waar de gebruiker deze rechtstreeks kan beheren. Lettertypesymbolen die niet geschikt zijn voor de flash.text.engine (FTE)-API's staan niet in de Bibliotheek en moeten dus handmatig worden beheerd. Met de volgende functie wordt een nieuw lettertype aan de Bibliotheek toegevoegd dat met de FTE-API's kan worden gebruikt.

```
function embedFontSymbol(symbolName, fontName, includeVariants) {
    var doc = fl.getDocumentDOM();
    if (doc) {
        // look up the item. if it exists, delete it.
        var index = doc.library.findItemIndex(symbolName);
        if (index > -1)
            doc.library.deleteItem(symbolName);

        // make a new font symbol in the library
        doc.library.addNewItem('font', symbolName);

        // look up the symbol by its name
        var index = doc.library.findItemIndex(symbolName);
        if (index > -1) {
            // get the item from the library and set the attributes of interest
            var fontObj = doc.library.items[index];
            fontObj.isDefineFont4Symbol = true;
            fontObj.font = fontName;
            fontObj.bold = false;
            fontObj.italic = false;
            fontObj.embedVariantGlyphs = includeVariants;
            // this is what forces the font into the SWF stream
            fontObj.linkageExportForAS = true;
            fontObj.linkageExportInFirstFrame = true;
        }
    }
}
```

Met de volgende functie worden alle lettertypesymbolen in het deelvenster Uitvoer weergegeven.

```
function dumpFontSymbols()
{
    var doc = fl.getDocumentDOM();
    if (doc) {
        var items = doc.library.items;
        fl.trace("items length = " + items.length);
        var i;
        for(i=0; i<items.length; i++) {
            var item = items[i];
            fl.trace("itemType = " + item.itemType);
            if (item.itemType == 'font') {
                fl.trace("name = " + item.name);
                fl.trace("DF4 symbol = " + item.isDefineFont4Symbol);
                fl.trace("font = " + item.font);
            }
        }
    }
}
```



### Zie ook

[fontItem.isDefineFont4Symbol](#), [text.embedVariantGlyphs](#)

## fontItem.font

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fontItem.font
```

### Beschrijving

Eigenschap; een tekenreeks die de naam opgeeft van het apparaatlettertype dat is gekoppeld aan het lettertype-item. Als u een tekenreeks invoert die niet overeenkomt met een geïnstalleerd apparaatlettertype, verschijnt er een foutbericht. Gebruik [fl.isFontInstalled\(\)](#) om na te gaan of er een lettertype aanwezig is op het systeem.

**Opmerking:** Als u deze waarde instelt, is de resulterende eigenschapswaarde mogelijk anders dan de tekenreeks die u invoert. Zie het volgende voorbeeld.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, wordt met de volgende code de naam weergegeven van het apparaatlettertype dat momenteel is gekoppeld aan het lettertype-item, en wordt dit vervolgens gewijzigd in Times:

```
fl.outputPanel.clear();
var theItem = fl.getDocumentDOM().library.items[0];
fl.trace(theItem.font);
theItem.font = "Times";
// depending on your system, the following may display something like "Times-Roman"
fl.trace(theItem.font);
```

## fontItem.isDefineFont4Symbol

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fontItem.isDefineFont4Symbol
```

### Beschrijving

Eigenschap; een Booleaanse waarde die de opmaak van het lettertype opgeeft dat bij publicatie van een SWF-bestand wordt uitgevoerd. Als deze waarde `true` is, voert Flash een lettertype uit dat geschikt is voor de `flash.text.engine` (FTE)-API's. Als deze waarde `false` is, is het lettertype geschikt voor de `flash.text`-API's, inclusief tekstvelden. De standaardwaarde is `false`.

### Voorbeeld

Zie `fontItem.embedVariantGlyphs`.

## fontItem.italic

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`fontItem.italic`

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of het lettertype-item cursief is (`true`) of niet (`false`).

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, wordt met de volgende code `true` weergegeven in het deelvenster Uitvoer als het item cursief is, en wordt `false` als dit niet zo is en wordt het lettertype-item vervolgens ingesteld op cursief:

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("italic: "+ theItem.italic);
theItem.italic=true;
fl.trace("italic: "+ theItem.italic);
```

## fontItem.size

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`fontItem.size`

### Beschrijving

Eigenschap; een geheel getal dat de grootte van het lettertype-item in punten vertegenwoordigt.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een lettertype-item is, wordt met de volgende code de puntgrootte van het item weergegeven in het deelvenster Uitvoer en wordt dit vervolgens ingesteld op 24.

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("font size: "+ theItem.size);
theItem.size=24;
fl.trace("font size: "+ theItem.size);
```

# Hoofdstuk 21: Object Frame

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Frame vertegenwoordigt frames in de laag.

## Overzicht van methoden

U kunt de volgende methoden gebruiken met het object Frame:

Methoden	Beschrijving
<code>frame.convertMotionObjectTo2D()</code>	Hiermee wordt het geselecteerde bewegingsobject geconverteerd naar een 2D-bewegingsobject.
<code>frame.convertMotionObjectTo3D()</code>	Hiermee wordt het geselecteerde bewegingsobject geconverteerd naar een 3D-bewegingsobject.
<code>frame.getCustomEase()</code>	Hiermee wordt een array van JavaScript-objecten geretourneerd, met elk een eigenschap <i>x</i> en <i>y</i> .
<code>frame.getMotionObjectXML()</code>	Hiermee wordt de bewegings-XML van het geselecteerde bewegingsobject geretourneerd.
<code>frame.hasMotionPath()</code>	Hiermee kunt u zien of de huidige selectie een bewegings-tween heeft of niet.
<code>frame.is3DMotionObject()</code>	Hiermee kunt u zien of de huidige selectie een 3D-bewegingsobject is of niet.
<code>frame.isMotionObject()</code>	Hiermee kunt u zien of de huidige selectie een bewegingsobject is of niet.
<code>frame.selectMotionPath()</code>	Hiermee wordt het bewegingspad van het huidige bewegingsobject geselecteerd of wordt de selectie hiervan ongedaan gemaakt.
<code>frame.setCustomEase()</code>	Hiermee wordt een cubische Bézier-curve aangegeven die als aangepaste versnellingsaanpassingscurve moet worden gebruikt.
<code>frame.setMotionObjectDuration()</code>	Hiermee wordt de duur (de lengte van de tween-reeks) opgegeven van het momenteel geselecteerde bewegingsobject.
<code>frame.setMotionObjectXML()</code>	Hiermee wordt de opgegeven bewegings-XML toegepast op het geselecteerde bewegingsobject.

## Overzicht van eigenschappen

U kunt de volgende eigenschappen gebruiken met het object Frame:

Eigenschap	Beschrijving
<code>frame.actionScript</code>	Een tekenreeks die ActionScript-code vertegenwoordigt.
<code>frame.duration</code>	Alleen-lezen; een geheel getal dat het aantal frames in een framereeks vertegenwoordigt.
<code>frame.elements</code>	Alleen-lezen; een array met objecten Element (zie <a href="#">Object Element</a> ).
<code>frame.hasCustomEase</code>	Een Booleaanse waarde die aangeeft of het frame zijn informatie over de versnellingsaanpassing uit de aangepaste versnellingsaanpassingscurve haalt.

Eigenschap	Beschrijving
<code>frame.labelType</code>	Een tekenreeks die het type van de framenaam aangeeft.
<code>frame.motionTweenOrientToPath</code>	Een Booleaanse waarde die aangeeft of het getweende element het element roteert terwijl het langs een pad beweegt om de hoek met elk punt op het pad aan te houden.
<code>frame.motionTweenRotate</code>	Een tekenreeks die aangeeft hoe het getweende element roteert.
<code>frame.motionTweenRotateTimes</code>	Een geheel getal dat aangeeft hoeveel keer het getweende element roteert tussen het eerste en het volgende hoofdfame.
<code>frame.motionTweenScale</code>	Een Booleaanse waarde die aangeeft of het getweende element zich in schaal aanpast aan de grootte van het object in het volgende hoofdfame, waarbij deze grootte in elk frame van de tween telkens toeneemt ( <code>true</code> ), of dat de schaal niet wordt aangepast ( <code>false</code> ).
<code>frame.motionTweenSnap</code>	Een Booleaanse waarde die aangeeft of het getweende element automatisch wordt uitgelijnd op het dichtstbijzijnde punt van de bij de laag van dit frame horende bewegingsgeleidelag ( <code>true</code> ), of niet ( <code>false</code> ).
<code>frame.motionTweenSync</code>	Een Booleaanse waarde die bij <code>true</code> de animatie van het getweende object synchroniseert met de hoofdtijdlijn.
<code>frame.name</code>	Een tekenreeks die de naam van het frame aangeeft.
<code>frame.shapeTweenBlend</code>	Een tekenreeks die aangeeft hoe een vorm-tween kan overvloeien tussen de vorm in het hoofdfame bij het begin van de tween en de vorm in het volgende hoofdfame.
<code>frame.soundEffect</code>	Een tekenreeks die effecten aangeeft voor een geluid dat rechtstreeks aan een frame is gekoppeld ( <code>frame.soundLibraryItem</code> ).
<code>frame.soundLibraryItem</code>	Een bibliotheekitem (zie <a href="#">Object SoundItem</a> ) dat wordt gebruikt om een geluid te maken.
<code>frame.soundLoop</code>	Een geheel getal dat aangeeft hoeveel keer een geluid dat rechtstreeks aan een frame is gekoppeld ( <code>frame.soundLibraryItem</code> ), wordt afgespeeld.
<code>frame.soundLoopMode</code>	Een tekenreeks die aangeeft of een geluid dat rechtstreeks aan een frame is gekoppeld ( <code>frame.soundLibraryItem</code> ), een bepaald aantal keren moet worden afgespeeld of continu in een lus moet worden herhaald.
<code>frame.soundName</code>	Een tekenreeks die de naam aangeeft voor een geluid dat rechtstreeks aan een frame is gekoppeld ( <code>frame.soundLibraryItem</code> ), zoals het in de bibliotheek staat opgeslagen.
<code>frame.soundSync</code>	Een tekenreeks die het sync-gedrag aangeeft voor een geluid dat rechtstreeks aan een frame is gekoppeld ( <code>frame.soundLibraryItem</code> ).
<code>frame.startFrame</code>	Alleen-lezen; de index van het eerste frame in een reeks.
<code>frame.tweenEasing</code>	Een geheel getal dat aangeeft hoeveel versnellingsaanpassing op het getweende project moet worden toegepast.
<code>frame.tweenInstanceName</code>	Hiermee wordt een instantienaam toegewezen aan het opgegeven bewegingsobject.
<code>frame.tweenType</code>	Een tekenreeks die het type van de tween aangeeft.
<code>frame.useSingleEaseCurve</code>	Een Booleaanse waarde die aangeeft of één aangepaste versnellingsaanpassingscurve wordt gebruikt om de informatie voor alle eigenschappen in snelheid aan te passen.

## frame.convertMotionObjectTo2D()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
frame.convertMotionObjectTo2D()
```

### Beschrijving

Methode; hiermee wordt het geselecteerde bewegingsobject geconverteerd naar een 2D-bewegingsobject.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde bewegingsobject geconverteerd naar een 2D-bewegingsobject:

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && the()) {
    theFrame.convertMotionObjectTo2D();
} else {
    fl.trace("It isn't motion or it's already a 2D motion");
}
```

## frame.convertMotionObjectTo3D()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
frame.convertMotionObjectTo3D()
```

### Beschrijving

Methode; hiermee wordt het geselecteerde bewegingsobject geconverteerd naar een 3D-bewegingsobject.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde bewegingsobject geconverteerd naar een 3D-bewegingsobject:

**Object Frame**

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && !theFrame.is3DMotionObject()){
theFrame.convertMotionObjectTo3D();
}else{
fl.trace("It isn't motion or it's already a 3D motion");
}
```

## frame.actionScript

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
frame.actionScript
```

**Beschrijving**

Eigenschap; een tekenreeks die ActionScript-code vertegenwoordigt. Gebruik "\n" om een nieuwe-lijn-teken in te voegen.

**Voorbeeld**

In het volgende voorbeeld wordt `stop()` aan de topaaghandeling van het eerste frame toegewezen:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].actionScript = 'stop()';
```

## frame.duration

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
frame.duration
```

**Beschrijving**

Alleen-lezen eigenschap; een geheel getal dat het aantal frames in een framereeks vertegenwoordigt.

**Voorbeeld**

In het volgende voorbeeld wordt het aantal frames opgeslagen in een framereeks, die begint te lopen vanaf het eerste frame in de topaag in de variabele `frameSpan`:

```
var frameSpan = fl.getDocumentDOM().getTimeline().layers[0].frames[0].duration;
```

## frame.elements

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.elements
```

### Beschrijving

Alleen-lezen eigenschap; een array van objecten `Element` (zie [Object Element](#)). De elementen staan in volgorde waarin ze zijn opgeslagen in het FLA-bestand. Als er in het werkgebied meerdere vormen los van elkaar staan, dan behandelt Flash ze als één element. Indien elke vorm in groep staat en er zich in het werkgebied dus meerdere groepen bevinden, dan beschouwt Flash ze als afzonderlijke elementen. Flash behandelt onbewerkte en niet-gegroepeerde vormen dus als één element, ongeacht het aantal afzonderlijke vorm in het werkgebied. Als er bijvoorbeeld drie onbewerkte, niet-gegroepeerde vormen in een frame staan, retourneert `elements.length` in dat frame een waarde 1. U kunt dit probleem omzeilen door elke vorm afzonderlijk te selecteren en te groeperen.

### Voorbeeld

In het volgende voorbeeld wordt een array met huidige elementen in de bovenste laag van het eerste frame opgeslagen in de variabele `myElements`:

```
var myElements = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
```

## frame.getCustomEase()

### Beschikbaarheid

Flash 8.

### Gebruik

```
Frame.getCustomEase([property])
```

### Parameters

**eigenschap** Een optionele tekenreeks die de eigenschap opgeeft waarvoor u de aangepaste versnelling wilt retourneren. Acceptabele waarden zijn "all", "position", "rotation", "scale", "color" en "filters". De standaardwaarde is "all".

### Geretourneerde waarde

Hiermee wordt een array van JavaScript-objecten geretourneerd, met elk een eigenschap *x* en *y*.

### Beschrijving

Methode; hiermee wordt een array van objecten geretourneerd die de controlepunten vertegenwoordigen voor de cubische Bézier-curve, die de versnellingsaanpassingscurve bepaalt.

### Voorbeeld

In het volgende voorbeeld wordt de aangepaste versnelling van de eigenschap `position` voor het eerste frame in de toplaag geretourneerd:

**Object Frame**

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
var easeArray = theFrame.getCustomEase("position");
```

**Zie ook**

[frame.hasCustomEase](#), [frame.setCustomEase\(\)](#), [frame.useSingleEaseCurve](#)

## frame.getMotionObjectXML()

**Beschikbaarheid**

Flash Professional CS5.

**Gebruik**

```
Frame.getMotionObjectXML()
```

**Beschrijving**

Hiermee wordt een tekenreeks geretourneerd van de bewegings-XML van het geselecteerde bewegingsobject.

**Voorbeeld**

In het volgende voorbeeld wordt de bewegings-XML geretourneerd van het geselecteerde bewegingsobject.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
//fl.trace(theFrame.getMotionObjectXML());
}else{
fl.trace("It is not motion.");
}
```

## frame.hasCustomEase

**Beschikbaarheid**

Flash 8.

**Gebruik**

```
frame.hasCustomEase
```

**Beschrijving**

Eigenschap; een Booleaanse waarde. Bij `true` haalt het frame zijn informatie over de versnellingsaanpassing uit de aangepaste versnellingsaanpassingscurve. Bij `false` haalt het frame zijn informatie over de versnellingsaanpassing uit de aangepaste versnelling.



**Object Frame****Voorbeeld**

In het volgende voorbeeld wordt aangegeven dat het eerste frame in de toplaag informatie beter uit de aangepaste versnelling kan halen dan uit de aangepaste versnellingsaanpassingscurve:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.hasCustomEase = false;
```

**Zie ook**

[frame.getCustomEase\(\)](#), [frame.setCustomEase\(\)](#), [frame.useSingleEaseCurve](#)

## frame.hasMotionPath()

**Beschikbaarheid**

Flash Professional CS5.

**Gebruik**

Frame.hasMotionPath()

**Beschrijving**

Methode; een Booleaanse waarde. Hiermee kunt u zien of de huidige selectie een bewegingspad bevat.

**Voorbeeld**

In het volgende voorbeeld wordt een traceringsinstructie geretourneerd, waarmee u kunt zien of de huidige selectie een bewegingspad bevat.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if (theFrame.isMotionObject()) {
    if (theFrame.hasMotionPath()) {
        fl.trace("There is a motion path");
    } else {
        fl.trace("There is no motion path");
    }
}
```

## frame.is3DMotionObject()

**Beschikbaarheid**

Flash Professional CS5.

**Gebruik**

Frame.is3DMotionObject()

### Beschrijving

Methode; een Booleaanse waarde. Hiermee kunt u zien of de huidige selectie een 3D-bewegingsobject is.

### Voorbeeld

In het volgende voorbeeld wordt een traceringsinstructie geretourneerd, waarmee u kunt zien of de huidige selectie een 3D-bewegingsobject is of niet.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && theFrame.is3DMotionObject()) {
    fl.trace("This selection is 3D Motion");
} else {
    fl.trace("This selection is not 3D motion");
}
```

## frame.isMotionObject()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
Frame.isMotionObject()
```

### Beschrijving

Methode; een Booleaanse waarde. Hiermee kunt u zien of de huidige selectie een bewegingsobject is.

### Voorbeeld

In het volgende voorbeeld wordt een traceringsinstructie geretourneerd, waarmee u kunt zien of de huidige selectie een bewegingsobject is of niet.

```
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
    var layer = my_tl.layers[my_tl.currentLayer];
    var frame = layer.frames[my_tl.currentFrame];
    return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
    fl.trace("This selection is motion.");
} else {
    fl.trace("This selection is not motion.");
}
```

## frame.labelType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`frame.labelType`

### Beschrijving

Eigenschap; een tekenreeks die het type van de framenaam aangeeft. Acceptabele waarden zijn "none", "name", "comment" en "anchor". Door een label in te stellen op "none", wordt de eigenschap `frame.name` gewist.

### Voorbeeld

In het volgende voorbeeld wordt de naam van het eerste frame in de toplaag ingesteld op "First Frame" en wordt het label vervolgens ingesteld op "comment":

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].labelType = 'comment';
```

## frame.motionTweenOrientToPath

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`frame.motionTweenOrientToPath`

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of het getweende element het element roteert terwijl het langs een pad beweegt om de hoek met elk punt op het pad aan te houden (`true`), of dat het niet roteert (`false`).

Als u een waarde voor deze eigenschap wilt opgeven, moet u `frame.motionTweenRotate` instellen op "none".

## frame.motionTweenRotate

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`frame.motionTweenRotate`

### Beschrijving

Eigenschap; een tekenreeks die aangeeft hoe het getweende element roteert. Acceptabele waarden zijn "none", "auto", "clockwise" en "counter-clockwise". De waarde "auto" betekent dat de richting waarin het object roteert een minimale beweging vereist om met de rotatie van het object in het volgende hoofdframe overeen te stemmen.

Als u een waarde voor `frame.motionTweenOrientToPath` wilt opgeven, moet u deze eigenschap instellen op "none".

### Voorbeeld

Zie `frame.motionTweenRotateTimes`.

## frame.motionTweenRotateTimes

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.motionTweenRotateTimes
```

### Beschrijving

Eigenschap; een geheel getal dat aangeeft hoeveel keer het getweende element roteert tussen het eerste en het volgende hoofdframe.

### Voorbeeld

In het volgende voorbeeld roteert het element in dit frame drie keer linksom voordat het het volgende hoofdframe bereikt:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotate = "counter-clockwise";  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotateTimes = 3;
```

## frame.motionTweenScale

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.motionTweenScale
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of het getweende element wordt geschaald naar de grootte van het object in het volgende hoofdframe, waarbij deze grootte in elk frame van de tween telkens toeneemt (`true`), of niet wordt geschaald (`false`).

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat het getweende element wordt geschaald naar de grootte van het object in het volgende hoofdfame, waardoor de grootte in elk frame van de tween toeneemt.

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenScale = true;
```

## frame.motionTweenSnap

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.motionTweenSnap
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of het getweende element automatisch wordt uitgelijnd op het dichtstbijzijnde punt van de bij de laag van dit frame horende bewegingsgeleidelaag (`true`), of niet (`false`).

## frame.motionTweenSync

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.motionTweenSync
```

### Beschrijving

Eigenschap; een Booleaanse waarde die bij `true` de animatie van het getweende object synchroniseert met de hoofdtijdlijn.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat het getweende object moet worden gesynchroniseerd met de tijdlijn:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenSync = true;
```

## frame.name

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.name
```

### Beschrijving

Eigenschap; een tekenreeks die de naam van het frame opgeeft.

### Voorbeeld

In het volgende voorbeeld wordt de naam van het eerste frame in de toplaag ingesteld op "First Frame", vervolgens wordt de waarde name opgeslagen in de variabele frameLabel:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';  
var frameLabel = fl.getDocumentDOM().getTimeline().layers[0].frames[0].name;
```

## frame.selectMotionPath()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
Frame.selectMotionPath()
```

### Beschrijving

Methode; een Booleaanse waarde. Hiermee wordt het bewegingspad van het huidige bewegingsobject geselecteerd (true) of wordt de selectie hiervan ongedaan gemaakt (false).

### Voorbeeld

In het voorbeeld wordt het bewegingspad van het huidige bewegingsobject geselecteerd of wordt de selectie hiervan ongedaan gemaakt.

```
var doc = fl.getDocumentDOM();  
var my_tl = doc.getTimeline();  
t  his.getCurrentFrame = function(){  
var layer = my_tl.layers[my_tl.c u rrentLayer];  
var frame = layer.frames[my_tl.currentFrame];  
return frame;  
}  
var theFrame = getCurrentFrame();  
if(theFrame.isMotionObject()){  
if (theFrame.hasMotionPath()){  
theFrame.selectMotionPath(true);  
}  
else{  
fl.trace("There is no motion path");  
}  
}else{  
fl.trace("It is no motion");  
}  
}
```

## frame.setCustomEase()

### Beschikbaarheid

Flash 8.

### Gebruik

```
frame.setCustomEase(property, easeCurve)
```

### Parameters

**eigenschap** Een tekenreeks die de eigenschap aangeeft waarvoor de versnellingsaanpassingscurve moet worden gebruikt. Acceptabele waarden zijn "all", "position", "rotation", "scale", "color" en "filters".

**easeCurve** Een array van objecten die de versnellingsaanpassingscurve bepaalt. Elk element van de array moet een JavaScript-object met *x*- en *y*- eigenschappen zijn.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een array van coördinaten opgegeven voor controlepunten en eindpunten van raaklijnen die een cubische Bézier-curve die moet worden gebruikt als aangepaste versnellingsaanpassingscurve. Deze array is opgebouwd uit de horizontale (ordinaal: links naar rechts) positie van de controlepunten en eindpunten van raaklijnen.

### Voorbeeld

In het volgende voorbeeld wordt de versnellingscurve voor alle eigenschappen van het eerste frame in de eerste laag ingesteld op de Bézier-curve die is opgegeven met de array `easeCurve`:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0];  
var easeCurve = [ {x:0,y:0}, {x:.3,y:.3}, {x:.7,y:.7}, {x:1,y:1} ];  
theFrame.setCustomEase( "all", easeCurve );
```

### Zie ook

[frame.getCustomEase\(\)](#), [frame.hasCustomEase](#), [frame.useSingleEaseCurve](#)

## frame.setMotionObjectDuration()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
Frame.setMotionObjectDuration( duration [, stretchExistingKeyframes] )
```

### Parameters

**duration** Hiermee wordt het aantal frames opgegeven voor de tween-reeks van het geselecteerde bewegingsobject.

**stretchExistingKeyframes** Een Booleaanse waarde die bepaalt of de tween-reeks wordt uitgerekt of dat er frames worden toegevoegd aan het einde van het laatste frame.

**Object Frame****Beschrijving**

Methode; hiermee wordt de duur (de lengte van de twee-reeks) ingesteld van het momenteel geselecteerde bewegingsobject.

**Voorbeeld**

In het volgende voorbeeld wordt een duur van elf frames opgegeven voor het geselecteerde bewegingsobject.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
theFrame.setMotionObjectDuration(11);
} else {
fl.trace("It isn't motion");
}
```

## frame.setMotionObjectXML()

**Beschikbaarheid**

Flash Professional CS5.

**Gebruik**

```
Frame.setMotionObjectXML( xmlstr [, endAtCurrentLocation] )
```

**Parameters**

**xmlstr** Een tekenreekswaarde die de XML-tekenreeks opgeeft.

**endAtCurrentLocation** Een Booleaanse waarde die bepaalt of de twee begint of eindigt op de huidige positie.

**Beschrijving**

Methode, hiermee wordt de opgegeven bewegings-XML toegepast op het geselecteerde bewegingsobject.

**Voorbeeld**

In dit voorbeeld wordt de bewegings-XML, die wordt aangeduid met `myMotionXML`, toegepast op het geselecteerde bewegingsobject.

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function() {
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
theFrame.setMotionObjectXML(myMotionXML.toString(), false);
```



## frame.shapeTweenBlend

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.shapeTweenBlend
```

### Beschrijving

Eigenschap; een tekenreeks die aangeeft hoe een vorm-tween kan overvloeien tussen de vorm in het hoofdframe bij het begin van de tween en de vorm in het volgende hoofdframe. Acceptabele waarden zijn "distributive" en "angular".

## frame.soundEffect

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.soundEffect
```

### Beschrijving

Eigenschap; een tekenreeks die effecten aangeeft voor een geluid dat rechtstreeks aan een frame is gekoppeld ([frame.soundLibraryItem](#)). Acceptabele waarden zijn "none", "left channel", "right channel", "fade left to right", "fade right to left", "fade in", "fade out" en "custom".

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat het aan het eerste frame gekoppelde geluid moet infaden:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundEffect = "fade in";
```

## frame.soundLibraryItem

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.soundLibraryItem
```

### Beschrijving

Eigenschap; een bibliotheekitem (zie [Object SoundItem](#)) dat wordt gebruikt om een geluid te maken. Het geluid is rechtstreeks aan het frame gekoppeld.

### Voorbeeld

In het volgende voorbeeld wordt het eerste item in de bibliotheek toegewezen aan de eigenschap `soundLibraryItem` van het eerste frame:

```
// The first item in the library must be a sound object.  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLibraryItem  
=fl.getDocumentDOM().library.items[0];
```

## frame.soundLoop

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`frame.soundLoop`

### Beschrijving

Eigenschap; een geheel getal dat aangeeft hoeveel keer een geluid dat rechtstreeks aan een frame is gekoppeld (`frame.soundLibraryItem`), wordt afgespeeld. Stel `frame.soundLoopMode` in op "repeat" als een waarde voor deze eigenschap wilt instellen.

### Voorbeeld

Zie `frame.soundLoopMode`.

## frame.soundLoopMode

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`frame.soundLoopMode`

### Beschrijving

Eigenschap; een tekenreeks die aangeeft of een geluid dat rechtstreeks aan een frame is gekoppeld (`frame.soundLibraryItem`) een bepaald aantal keren moet worden afgespeeld of continu in een lus moet worden herhaald. Acceptabele waarden zijn "repeat" en "loop". Stel een waarde in voor `frame.soundLoop` om aan te geven hoeveel keer het geluid moet worden afgespeeld.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat het geluid tweemaal moet worden afgespeeld:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoopMode = "repeat";  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoop = 2;
```

## frame.soundName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.soundName
```

### Beschrijving

Eigenschap; een tekenreeks die de naam aangeeft voor een geluid dat rechtstreeks aan een frame is gekoppeld ([frame.soundLibraryItem](#)), zoals het in de bibliotheek staat opgeslagen.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `soundName` van het eerste frame gewijzigd in `"song1.mp3"`; `song1.mp3` moet in de bibliotheek staan:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundName = "song1.mp3";
```

## frame.soundSync

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.soundSync
```

### Beschrijving

Eigenschap; een tekenreeks die het sync-gedrag aangeeft voor een geluid dat rechtstreeks aan een frame is gekoppeld ([frame.soundLibraryItem](#)). Acceptabele waarden zijn "event", "stop", "start" en "stream".

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat een geluid moet streamen:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundSync = 'stream';
```

## frame.startFrame

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.startFrame
```

### Beschrijving

Alleen-lezen eigenschap; de index van het eerste frame in een reeks.

### Voorbeeld

In het volgende voorbeeld is `stFrame` de index van het eerste frame in de framereeks. In dit voorbeeld omvat een framereeks de zes frames van Frame 5 tot en met Frame 10. Tussen Frame 5 en Frame 10 is de waarde van `stFrame` dan ook constant 4 (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummerwaarden).

```
var stFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[4].startFrame;  
fl.trace(stFrame); // 4  
var stFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[9].startFrame;  
fl.trace(stFrame); // 4
```

## frame.tweenEasing

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
frame.tweenEasing
```

### Beschrijving

Eigenschap; een geheel getal dat aangeeft hoeveel versnellingsaanpassing op het getweende project moet worden toegepast. Acceptabele waarden liggen tussen -100 en 100. Wanneer u wilt dat de bewegings-tween langzaam begint en versnelt aan het einde van de animatie, moet u een waarde tussen -1 en -100 gebruiken. Wanneer u wilt dat de bewegings-tween snel begint en vertraagt aan het einde van de animatie, moet u een positieve waarde tussen 1 en 100 gebruiken.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat de beweging van het getweende object behoorlijk snel moet beginnen en aan het einde van de animatie moet vertragen:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenEasing = 50;
```

## frame.tweenInstanceName

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
Frame.tweenInstanceName()
```

### Beschrijving

Eigenschap; een tekenreeks die een instantienaam toewijst aan het geselecteerde bewegingsobject.

### Voorbeeld

In het volgende voorbeeld wordt de instantienaam `MyMotionTween` aan het opgegeven bewegingsobject toegevoegd.

```
theFrame.tweenInstanceName = "MyMotionTween";
```

## frame.tweenType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`frame.tweenType`

### Beschrijving

Eigenschap; een tekenreeks die het type van de tween aangeeft; acceptabele waarden zijn "motion", "shape" en "none". Met de waarde "none" wordt de bewegings-tween verwijderd. Gebruik de methode `timeline.createMotionTween()` om een bewegings-tween te maken.

Wanneer u "motion" aangeeft, moet het object in het frame een symbool, tekstveld of gegroepeerd object zijn. Het wordt vanaf de locatie in het huidige hoofdframe naar de locatie in het volgende hoofdframe getweend.

Wanneer u "shape" aangeeft, moet het object in het frame een vorm zijn. Het vloeit vanuit de vorm in het huidige hoofdframe over naar de vorm in het volgende hoofdframe.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat het object een bewegings-tween is. Het moet daarom vanaf de locatie in het huidige hoofdframe worden getweend naar de locatie in het volgende hoofdframe:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenType = "motion";
```

## frame.useSingleEaseCurve

### Beschikbaarheid

Flash 8.

### Gebruik

`frame.useSingleEaseCurve`

### Beschrijving

Eigenschap; een Booleaanse waarde. Bij `true` wordt één aangepaste versnellingsaanpassingscurve gebruikt om de informatie voor alle eigenschappen in snelheid aan te passen. Bij `false` heeft elke eigenschap haar eigen versnellingsaanpassingscurve.

Deze eigenschap wordt genegeerd als de aangepaste versnellingsaanpassing in het frame niet is geactiveerd.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat voor alle eigenschappen van het eerste frame op de eerste laag één aangepaste versnellingsaanpassingscurve moet worden gebruikt:

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]  
theFrame.useSingleEaseCurve = true;
```

### Zie ook

`frame.getCustomEase()`, `frame.hasCustomEase`, `frame.setCustomEase()`

# Hoofdstuk 22: Object HalfEdge

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object HalfEdge is de geleide zijde van de rand van een [Object Shape](#). Een rand heeft twee halve randen. U kunt de contouren van een vorm transversaal doorlopen door rond deze halve randen te "wandelen". U begint bijvoorbeeld bij een halve rand en kunt alle halve randen rond een contour van een vorm traceren en terugkeren naar de oorspronkelijke halve rand.

Halve randen zijn geordend. Een halve rand vertegenwoordigt een zijde van de rand; de andere halve rand vertegenwoordigt de andere zijde.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object HalfEdge:

Methode	Beschrijving
<code>halfEdge.getEdge()</code>	Hiermee wordt het <a href="#">Object Edge</a> opgehaald voor het object HalfEdge.
<code>halfEdge.getNext()</code>	Hiermee wordt de volgende halve rand opgehaald van de huidige contour.
<code>halfEdge.getOppositeHalfEdge()</code>	Hiermee wordt het object HalfEdge opgehaald aan de andere zijde van de rand.
<code>halfEdge.getPrev()</code>	Hiermee wordt het voorgaande object HalfEdge opgehaald van de huidige contour.
<code>halfEdge.getVertex()</code>	Hiermee wordt het <a href="#">Object Vertex</a> opgehaald van de kop van het object HalfEdge.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object HalfEdge:

Eigenschap	Beschrijving
<code>halfEdge.id</code>	Alleen-lezen; een uniek geheel getal als id voor het object HalfEdge.
<code>halfEdge.index</code>	Een geheel getal met de waarde 0 of 1 dat de index voor dit object HalfEdge opgeeft in de bovenliggende rand.

## halfEdge.getEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
halfEdge.getEdge()
```

### Parameters

Geen.

### Geretourneerde waarde

Een [Object Edge](#).

### Beschrijving

Methode; hiermee wordt het object Edge opgehaald voor het object HalfEdge. Zie [Object Edge](#).

### Voorbeeld

In het volgende voorbeeld wordt gedemonstreerd hoe een rand en een halve rand worden opgehaald voor de opgegeven vorm:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var edge = hEdge.getEdge();
```

## halfEdge.getNext()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
halfEdge.getNext()
```

### Parameters

Geen.

### Geretourneerde waarde

Een object HalfEdge.

### Beschrijving

Methode; hiermee wordt de volgende halve rand opgehaald voor de huidige contour.

**Opmerking:** Hoewel halve randen een richting en volgorde hebben, hebben randen die niet.

### Voorbeeld

In het volgende voorbeeld wordt de eerste halve rand van de opgegeven contour opgeslagen in de variabele nextHalfEdge:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge( 0 );  
var nextHalfEdge = hEdge.getNext();
```

## halfEdge.getOppositeHalfEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
halfEdge.getOppositeHalfEdge()
```

### Parameters

Geen.

### Geretourneerde waarde

Een object HalfEdge.

### Beschrijving

Methode; hiermee wordt het object HalfEdge opgehaald aan de andere zijde van de rand.

### Voorbeeld

In het volgende voorbeeld wordt de halve rand tegenover hEdge opgeslagen in de variabele otherHalfEdge:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var otherHalfEdge = hEdge.getOppositeHalfEdge();
```

## halfEdge.getPrev()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
halfEdge.getPrev()
```

### Parameters

Geen.

### Geretourneerde waarde

Een object HalfEdge.

### Beschrijving

Methode; hiermee wordt het voorgaande object HalfEdge opgehaald van de huidige contour.

**Opmerking:** Hoewel halve randen een richting en volgorde hebben, hebben randen die niet.

### Voorbeeld

In het volgende voorbeeld wordt de vorige halve rand van de opgegeven contour opgeslagen in de variabele prevHalfEdge:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge( 0 );  
var prevHalfEdge = hEdge.getPrev();
```



## halfEdge.getVertex()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
halfEdge.getVertex()
```

### Parameters

Geen.

### Geretourneerde waarde

Een [Object Vertex](#)

### Beschrijving

Methode; hiermee wordt het object Vertex opgehaald van de kop van het object HalfEdge. Zie [Object Vertex](#)

### Voorbeeld

In het volgende voorbeeld wordt het object Vertex op de kop van hEdge opgeslagen in de variabele vertex:

```
var shape = fl.getDocumentDOM().selection[0];  
var edge = shape.edges[0];  
var hEdge = edge.getHalfEdge(0);  
var vertex = hEdge.getVertex();
```

## halfEdge.id

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
halfEdge.id
```

### Beschrijving

Alleen-lezen eigenschap; een uniek geheel getal als id voor het object HalfEdge.

### Voorbeeld

In het volgende voorbeeld wordt een unieke id voor de opgegeven halve rand weergegeven in het deelvenster Uitvoer:

```
var shape = fl.getDocumentDOM().selection[0];  
alert(shape.contours[0].getHalfEdge().id);
```

## halfEdge.index

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`halfEdge.index`

### Beschrijving

Alleen-lezen eigenschap; een geheel getal met de waarde 0 of 1 dat de index voor dit object HalfEdge opgeeft in de bovenliggende rand.

### Voorbeeld

In het volgende voorbeeld wordt de indexwaarde voor de opgegeven halve rand weergegeven in het deelvenster Uitvoer:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var heIndex = hEdge.index;
```

# Hoofdstuk 23: Object Instance

**Overerving** [Object Element](#) > Object Instance

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Instance is een subklasse van het [Object Element](#).

## Overzicht van eigenschappen

Naast alle eigenschappen van het object Element heeft Instance de volgende eigenschappen:

Eigenschap	Beschrijving
<a href="#">instance.instanceType</a>	Alleen-lezen; een tekenreeks die het type instantie vertegenwoordigt.
<a href="#">instance.libraryItem</a>	Bibliotheekitem dat wordt gebruikt om deze instantie te instantiëren.

## instance.instanceType

### Beschikbaarheid

Flash MX 2004; mogelijke waarde "video" toegevoegd in Flash 8.

### Gebruik

```
instance.instanceType
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het type instantie vertegenwoordigt. Mogelijke waarden zijn "symbol", "bitmap", "embedded video", "linked video", "video" en "compiled clip".

In Flash MX 2004 is "embedded\_video" de waarde van `instance.instanceType` voor een item dat is toegevoegd aan de bibliotheek met behulp van `library.addNewItem("video")`. In Flash 8 en hoger is de waarde "video". Zie [library.addNewItem\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt gedemonstreerd dat het type instantie van een filmclip `symbol` is:

```
// Select a movie clip and then run this script.
var type = fl.getDocumentDOM().selection[0].instanceType;
fl.trace("This instance type is " + type);
```

## instance.libraryItem

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`instance.libraryItem`

### Beschrijving

Eigenschap; een bibliotheekitem dat wordt gebruikt om deze instantie te instantiëren. U kunt deze eigenschap alleen wijzigen in een ander bibliotheekitem van hetzelfde type (u kunt dus niet een instantie van `symbol` instellen om te verwijzen naar een bitmap). Zie [Object library](#).

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde symbool gewijzigd waarmee wordt verwezen naar het eerste item in de bibliotheek:

```
f1.getDocumentDOM().selection[0].libraryItem = f1.getDocumentDOM().library.items[0];
```

# Hoofdstuk 24: Object Item

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Item is een abstracte basisklasse. Alle elementen in de bibliotheek zijn afgeleid van Item. Zie ook [Object library](#).

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Item:

Method	Beschrijving
<code>item.addData()</code>	Hiermee worden opgegeven gegevens toegevoegd aan een bibliotheekitem.
<code>item.getData()</code>	Hiermee wordt de waarde van de opgegeven gegevens opgehaald.
<code>item.hasData()</code>	Hiermee wordt bepaald of het bibliotheekitem de benoemde gegevens heeft.
<code>item.removeData()</code>	Hiermee worden permanente gegevens verwijderd uit het bibliotheekitem.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Item:

Eigenschap	Beschrijving
<code>item.itemType</code>	Alleen-lezen; een tekenreeks die het type element opgeeft.
<code>item.linkageBaseClass</code>	Een tekenreeks die de ActionScript 3.0-klasse opgeeft die aan het symbool zal worden gekoppeld.
<code>item.linkageClassName</code>	Een tekenreeks die de ActionScript 2.0-klasse opgeeft die aan het symbool zal worden gekoppeld.
<code>item.linkageExportForAS</code>	Een Booleaanse waarde. Als de waarde <code>true</code> is, wordt het item geëxporteerd voor ActionScript.
<code>item.linkageExportForRS</code>	Een Booleaanse waarde. Als de waarde <code>true</code> is, wordt het item geëxporteerd voor het delen tijdens runtime.
<code>item.linkageExportInFirstFrame</code>	Een Booleaanse waarde. Als de waarde <code>true</code> is, wordt het item geëxporteerd in het eerste frame.
<code>item.linkageIdentifier</code>	Een tekenreeks die de naam opgeeft die Flash gebruikt om het element te identificeren wanneer het aan het doel-SWF-bestand wordt gekoppeld.
<code>item.linkageImportForRS</code>	Een Booleaanse waarde. Als de waarde <code>true</code> is, wordt het item geïmporteerd voor het delen tijdens runtime.
<code>item.linkageURL</code>	Een tekenreeks die de URL opgeeft waar het SWF-bestand dat het gedeelde element bevat, zich bevindt.
<code>item.name</code>	Een tekenreeks die de naam opgeeft van het bibliotheekitem dat de mapstructuur bevat.

## item.addData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.addData(name, type, data)
```

### Parameters

**name** Een tekenreeks die de naam van de gegevens opgeeft.

**type** Een tekenreeks die het type van de gegevens opgeeft. Geldige typen zijn "integer", "integerArray", "double", "doubleArray", "string" en "byteArray".

**data** De gegevens die moeten worden toegevoegd aan het opgegeven bibliotheekitem. Het type gegevens is afhankelijk van de waarde van de parameter type. Als het type bijvoorbeeld "integer" is, moet de waarde van de gegevens een geheel getal zijn, enzovoort.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden opgegeven gegevens toegevoegd aan een bibliotheekitem.

### Voorbeeld

In het volgende voorbeeld worden gegevens met de naam `myData` en met als waarde het gehele getal 12 toegevoegd aan het eerste item in de bibliotheek:

```
fl.getDocumentDOM().library.items[0].addData("myData", "integer", 12);
```

## item.getData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.getData(name)
```

### Parameters

**name** Een tekenreeks waarmee de naam wordt opgegeven van de gegevens die moet worden opgehaald.

### Geretourneerde waarde

De gegevens die worden opgegeven door de parameter *name*. Het type geretourneerde gegevens is afhankelijk van het type opgeslagen gegevens.

### Beschrijving

Methode; hiermee wordt de waarde van de opgegeven gegevens opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de gegevens met de naam `myData` opgehaald van het eerste item in de bibliotheek en opgeslagen in de variabele `libData`:

```
var libData = fl.getDocumentDOM().library.items[0].getData("myData");
```

## item.hasData()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.hasData (name)
```

### Parameters

**name** Een tekenreeks waarmee de naam wordt opgegeven van de gegevens waarop moet worden gecontroleerd in het bibliotheekitem.

### Geretoureerde waarde

Een Booleaanse waarde: `true` als de opgegeven gegevens bestaan, anders `false`.

### Beschrijving

Methode; hiermee wordt bepaald of het bibliotheekitem de benoemde gegevens heeft.

### Voorbeeld

In het volgende voorbeeld wordt een bericht getoond in het deelvenster Uitvoer als het eerste item in de bibliotheek het gegevenspunt met de naam `myData` bevat:

```
if (fl.getDocumentDOM().library.items[0].hasData("myData")) {  
    fl.trace("Yep, it's there!");  
}
```

## item.itemType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.itemType
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het type element opgeeft. Mogelijke waarden zijn: "undefined", "component", "movie clip", "graphic", "button", "folder", "font", "sound", "bitmap", "compiled clip", "screen" en "video". Als deze eigenschap "video" is, kunt u het type video bepalen; zie [videoItem.videoType](#).

### Voorbeeld

In het volgende voorbeeld wordt het type van het opgegeven bibliotheekitem getoond in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().library.items[0].itemType);
```

## item.linkageBaseClass

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
item.linkageBaseClass
```

### Beschrijving

Eigenschap; een tekenreeks die de ActionScript 3.0-klasse opgeeft die aan het symbool zal worden gekoppeld. De waarde die hier wordt opgegeven, wordt weergegeven in het dialoogvenster Koppeling in de ontwerpomgeving, en in andere dialoogvensters die besturingselementen voor het dialoogvenster Koppeling bevatten, zoals het dialoogvenster Symbooleigenschappen. (Gebruik [item.linkageClassName](#) om deze waarde op te geven voor een ActionScript 2.0-klasse.)

Als de basisklasse de standaardwaarde is voor het type symbool (bijvoorbeeld "flash.display.MovieClip" voor filmclips, "flash.display.SimpleButton" voor knoppen enzovoort), is deze eigenschap een lege tekenreeks (""). Als u een item de standaardbasisklasse wilt maken, stelt u deze waarde ook in op een lege tekenreeks.

Wanneer u deze waarde instelt, worden de controles die worden uitgevoerd door het dialoogvenster Koppeling, niet uitgevoerd. Bovendien worden er geen fouten gegenereerd als Flash niet in staat is de basisklasse in te stellen op de opgegeven waarde. Als u deze waarde bijvoorbeeld instelt in het dialoogvenster Koppeling, worden controles afgedwongen om te garanderen dat de basisklasse kan worden gevonden in het klassepap van het FLA-bestand. Zo weet u zeker dat ActionScript 3.0 wordt gekozen op het tabblad Flash van het dialoogvenster Publicatie-instellingen, enzovoort. Deze controles worden niet uitgevoerd als u deze eigenschap instelt in een script.

### Voorbeeld

In de volgende regels met code worden enkele manieren getoond om deze eigenschap te gebruiken:

```
// sets the library item base class to "Sprite"
fl.getDocumentDOM().library.items[0].linkageBaseClass = "flash.display.Sprite";
// sets the library item base class to the default for that item type
fl.getDocumentDOM().library.items[0].linkageBaseClass = "";
// finds and displays the library item's base class
fl.trace(fl.getDocumentDOM().library.items[0].linkageBaseClass);
```

### Zie ook

[document.docClass](#)

## item.linkageClassName

### Beschikbaarheid

Flash MX 2004.



### Gebruik

`item.linkageClassName`

### Beschrijving

Eigenschap; een tekenreeks die de ActionScript 2.0-klasse opgeeft die aan het symbool zal worden gekoppeld. (Gebruik `item.linkageBaseClass` om deze waarde op te geven voor een ActionScript 3.0-klasse.)

Deze eigenschap kan alleen worden gedefinieerd als de eigenschap `item.linkageExportForAS` en/of `item.linkageExportForRS` is ingesteld op `true`, en als de eigenschap `item.linkageImportForRS` is ingesteld op `false`.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat de naam van de ActionScript 2.0-klasse die wordt gekoppeld aan het eerste item in de bibliotheek, `myClass` is:

```
fl.getDocumentDOM().library.items[0].linkageClassName = "myClass";
```

## item.linkageExportForAS

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`item.linkageExportForAS`

### Beschrijving

Eigenschap; een Booleaanse waarde. Als deze eigenschap `true` is, wordt het item geëxporteerd voor ActionScript. U kunt ook de eigenschappen `item.linkageExportForRS` en `item.linkageExportInFirstFrame` instellen op `true`.

Als u instelt dat deze eigenschap "true" is, moet de eigenschap `item.linkageImportForRS` worden ingesteld op `false`. Daarnaast moet u een id (`item.linkageIdentifier`) en een URL (`item.linkageURL`) opgeven.

### Voorbeeld

In het volgende voorbeeld wordt deze eigenschap ingesteld voor het opgegeven bibliotheekitem:

```
fl.getDocumentDOM().library.items[0].linkageExportForAS = true;
```

## item.linkageExportForRS

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`item.linkageExportForRS`

### Beschrijving

Eigenschap; een Booleaanse waarde. Als deze eigenschap `true` is, wordt het item geëxporteerd voor het delen tijdens runtime. U kunt ook de eigenschappen `item.linkageExportForAS` en `item.linkageExportInFirstFrame` instellen op `true`.

Als u instelt dat deze eigenschap "true" is, moet de eigenschap `item.linkageImportForRS` worden ingesteld op `false`. Daarnaast moet u een id (`item.linkageIdentifier`) en een URL (`item.linkageURL`) opgeven.

### Voorbeeld

In het volgende voorbeeld wordt deze eigenschap ingesteld voor het opgegeven bibliotheekitem:

```
fl.getDocumentDOM().library.items[0].linkageExportForRS = true;
```

## item.linkageExportInFirstFrame

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.linkageExportInFirstFrame
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Als de waarde `true` is, wordt het item geëxporteerd naar het eerste frame; als de waarde `false` is, wordt het item geëxporteerd naar het frame van de eerste instantie. Als het item niet wordt weergegeven op het werkgebied, wordt het niet geëxporteerd.

Deze eigenschap kan alleen worden ingesteld op `true` als `item.linkageExportForAS` en/of `item.linkageExportForRS` zijn ingesteld op `true`.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat het opgegeven bibliotheekitem wordt geëxporteerd in het eerste frame.

```
fl.getDocumentDOM().library.items[0].linkageExportInFirstFrame = true;
```

## item.linkageIdentifier

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.linkageIdentifier
```

### Beschrijving

Eigenschap; een tekenreeks die de naam opgeeft die Flash gebruikt om het element te identificeren wanneer het aan het doel-SWF-bestand wordt gekoppeld. Flash negeert deze eigenschap als `item.linkageImportForRS`, `item.linkageExportForAS` en `item.linkageExportForRS` zijn ingesteld op `false`. Omgekeerd moet deze eigenschap worden ingesteld als een van deze eigenschappen is ingesteld op `true`.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat de tekenreeks `my_mc` wordt gebruikt om het bibliotheekitem te identificeren wanneer het wordt gekoppeld aan het doel-SWF-bestand waarnaar het wordt geëxporteerd:

```
fl.getDocumentDOM().library.items[0].linkageIdentifier = "my_mc";
```

### Zie ook

[item.linkageURL](#)

## item.linkageImportForRS

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.linkageImportForRS
```

### Beschrijving

Eigenschap; een Booleaanse waarde: als de waarde `true` is, wordt het item geïmporteerd voor het delen tijdens runtime. Als deze eigenschap is ingesteld op `true`, moeten zowel [item.linkageExportForAS](#) als [item.linkageExportForRS](#) zijn ingesteld op `false`. Daarnaast moet u een id ([item.linkageIdentifier](#)) en een URL ([item.linkageURL](#)) opgeven.

### Voorbeeld

In het volgende voorbeeld wordt deze eigenschap ingesteld op `true` voor het opgegeven bibliotheekitem:

```
fl.getDocumentDOM().library.items[0].linkageImportForRS = true;
```

## item.linkageURL

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
item.linkageURL
```

### Beschrijving

Eigenschap; een tekenreeks die de URL opgeeft waar het SWF-bestand dat het gedeelde element bevat, zich bevindt. Flash negeert deze eigenschap als [item.linkageImportForRS](#), [item.linkageExportForAS](#) en [item.linkageExportForRS](#) zijn ingesteld op `false`. Omgekeerd moet deze eigenschap worden ingesteld als een van deze eigenschappen is ingesteld op `true`. U kunt een web-URL of een bestandsnaam opgeven in een platformafhankelijke notatie (dat wil zeggen slashes [/] of backslashes [\], afhankelijk van het platform).

### Voorbeeld

In het volgende voorbeeld wordt een koppelings-URL opgegeven voor het opgegeven bibliotheekitem:

```
fl.getDocumentDOM().library.items[0].linkageURL = "theShareSWF.swf";
```

#### Zie ook

[item.linkageIdentifier](#)

## item.name

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

`item.name`

#### Beschrijving

Methode; een tekenreeks die de naam opgeeft van het bibliotheekitem dat de mapstructuur bevat. Als bijvoorbeeld `Symbol_1` in een map staat met de naam `Map_1`, is de eigenschap `name` van `Symbol_1` "`Map_1/Symbol_1`".

#### Voorbeeld

In het volgende voorbeeld wordt de naam van het opgegeven bibliotheekitem getoond in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().library.items[0].name);
```

## item.removeData()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

`item.removeData(name)`

#### Parameters

**name** Hiermee wordt de naam opgegeven van de gegevens die moeten worden verwijderd uit het bibliotheekitem.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Eigenschap; hiermee worden permanente gegevens verwijderd uit het bibliotheekitem.

#### Voorbeeld

In het volgende voorbeeld worden de gegevens met de naam `myData` verwijderd uit het eerste item in de bibliotheek:

```
fl.getDocumentDOM().library.items[0].removeData("myData");
```

# Hoofdstuk 25: Object Layer

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Layer vertegenwoordigt een laag in de tijdlijn. De eigenschap `timeline.layers` bevat een array van objecten Layer die kunnen worden benaderd met `fl.getDocumentDOM().getTimeline().layers`.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Layer:

Eigenschap	Beschrijving
<code>layer.color</code>	Een tekenreeks, hexadecimale waarde of geheel getal die of dat de kleur opgeeft die is toegewezen aan de contour van de laag.
<code>layer.frameCount</code>	Alleen-lezen; een geheel getal dat het aantal frames in de laag opgeeft.
<code>layer.frames</code>	Alleen-lezen; array van objecten Frame.
<code>layer.height</code>	Een geheel getal dat het percentage van de laaghoogte opgeeft; is gelijk aan de waarde voor Laaghoogte in het dialoogvenster Laageigenschappen.
<code>layer.layerType</code>	Een tekenreeks die het huidige gebruik van de laag opgeeft; is gelijk aan de instelling Type in het dialoogvenster Laageigenschappen.
<code>layer.locked</code>	Een Booleaanse waarde die de vergrendelingsstatus van de laag opgeeft.
<code>layer.name</code>	Een tekenreeks die de naam van de laag opgeeft.
<code>layer.outline</code>	Een Booleaanse waarde die de status van contouren voor alle objecten in de laag opgeeft.
<code>layer.parentLayer</code>	Een object Layer dat de bovenliggende map, hulplijnlaag of maskeerlaag vertegenwoordigt.
<code>layer.visible</code>	Een Booleaanse waarde die opgeeft of de objecten van de laag in het werkgebied worden weergegeven of verborgen.

## layer.color

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`layer.color`

### Beschrijving

Eigenschap; de kleur die is toegewezen aan de contour van de laag, in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB

- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

Deze eigenschap is gelijk aan de instelling Contourkleur in het dialoogvenster Laageigenschappen.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eerste laag opgeslagen in de variabele `colorValue`:

```
var colorValue = fl.getDocumentDOM().getTimeline().layers[0].color;
```

In het volgende voorbeeld worden drie manieren getoond om in te stellen dat de kleur van de eerste laag rood is:

```
fl.getDocumentDOM().getTimeline().layers[0].color=16711680;  
fl.getDocumentDOM().getTimeline().layers[0].color="#ff0000";  
fl.getDocumentDOM().getTimeline().layers[0].color=0xFF0000;
```

## layer.frameCount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.frameCount
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat het aantal frames in de laag opgeeft.

### Voorbeeld

In het volgende voorbeeld wordt het aantal frames in de eerste laag opgeslagen in de variabele `fcNum`:

```
var fcNum = fl.getDocumentDOM().getTimeline().layers[0].frameCount;
```

## layer.frames

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.frames
```

### Beschrijving

Alleen-lezen eigenschap; een array van objecten `Frame` (zie [Object Frame](#)).

### Voorbeeld

In het volgende voorbeeld wordt de variabele eigenschap `frameArray` ingesteld op het array van objecten `Frame` voor de frames in het huidige document:

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
```

Als u wilt bepalen of een frame een hoofdframe is, controleert u of de eigenschap `frame.startFrame` overeenkomt met de matrixindex, zoals getoond in het volgende voorbeeld:

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
var n = frameArray.length;
for (i=0; i<n; i++) {
    if (i==frameArray[i].startFrame) {
        alert("Keyframe at: " + i);
    }
}
```

## layer.height

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`layer.height`

### Beschrijving

Eigenschap; een geheel getal dat het percentage van de laaghoogte opgeeft; is gelijk aan de waarde voor Laaghoogte in het dialoogvenster Laageigenschappen. Acceptabele waarden zijn percentages van de standaardhoogte: 100, 200 of 300.

### Voorbeeld

In het volgende voorbeeld wordt de percentagewaarde van de ingestelde hoogte van de eerste laag opgeslagen:

```
var layerHeight = fl.getDocumentDOM().getTimeline().layers[0].height;
```

In het volgende voorbeeld wordt de hoogte van de eerste laag ingesteld op 300 procent:

```
fl.getDocumentDOM().getTimeline().layers[0].height = 300;
```

## layer.layerType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`layer.layerType`

### Beschrijving

Eigenschap; een tekenreeks die het huidige gebruik van de laag opgeeft; is gelijk aan de instelling Type in het dialoogvenster Laageigenschappen. Acceptabele waarden zijn "normal", "guide", "guided", "mask", "masked" en "folder".

### Voorbeeld

In het volgende voorbeeld wordt de eerste laag in de tijdlijn ingesteld op het type folder:

```
fl.getDocumentDOM().getTimeline().layers[0].layerType = "folder";
```

## layer.locked

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.locked
```

### Beschrijving

Eigenschap; een Booleaanse waarde die de vergrendelingsstatus van de laag opgeeft. Als de waarde `true` is, is de laag vergrendeld. De standaardwaarde is `false`.

### Voorbeeld

In het volgende voorbeeld wordt de Booleaanse waarde voor de status van de eerste laag opgeslagen in de variabele `lockStatus`:

```
var lockStatus = fl.getDocumentDOM().getTimeline().layers[0].locked;
```

In het volgende voorbeeld wordt de status van de eerste laag ingesteld op niet-vergrendeld:

```
fl.getDocumentDOM().getTimeline().layers[0].locked = false;
```

## layer.name

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.name
```

### Beschrijving

Eigenschap; een tekenreeks die de naam van de laag opgeeft.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de eerste laag in het huidige document ingesteld op `foreground`:

```
fl.getDocumentDOM().getTimeline().layers[0].name = "foreground";
```

## layer.outline

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.outline
```



### Beschrijving

Eigenschap; een Booleaanse waarde die de status van contouren voor alle objecten in de laag opgeeft. Als de waarde `true` is, worden alle objecten in de laag alleen zonder contouren weergegeven. Als de waarde `false` is, worden objecten weergegeven zoals ze zijn gemaakt.

### Voorbeeld

In het volgende voorbeeld wordt ingesteld dat alle objecten in de eerste laag alleen zonder contouren worden weergegeven:

```
fl.getDocumentDOM().getTimeline().layers[0].outline = true;
```

## layer.parentLayer

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.parentLayer
```

### Beschrijving

Eigenschap; een object Layer dat de bovenliggende map, hulplijnlaag of maskeerlaag vertegenwoordigt. De bovenliggende laag moet een map, hulplijn- of maskeerlaag zijn die voorafgaat aan de laag, of de `parentLayer` van de voorafgaande of volgende laag zijn. Door het instellen van de `parentLayer` van een laag, verandert de positie van de laag in de lijst niet. Als u probeert de `parentLayer` van een laag in te stellen op een laag die hiervoor moeten worden verplaatst, heeft dit geen effect. Voor een laag op hoofdniveau wordt `null` gebruikt.

### Voorbeeld

In het volgende voorbeeld worden twee lagen gebruikt op hetzelfde niveau op dezelfde tijdlijn. De eerste laag (`layers[0]`) wordt omgezet in een map en vervolgens ingesteld als bovenliggende map van de tweede laag (`layers[1]`). Door deze handeling wordt de tweede laag in de eerste laag geplaatst.

```
var parLayer = fl.getDocumentDOM().getTimeline().layers[0];  
parLayer.layerType = "folder";  
fl.getDocumentDOM().getTimeline().layers[1].parentLayer = parLayer;
```

## layer.visible

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
layer.visible
```

### **Beschrijving**

Eigenschap; een Booleaanse waarde die opgeeft of de objecten van de laag in het werkgebied worden weergegeven of verborgen. Als de waarde `true` is, zijn alle objecten in de laag zichtbaar; als de waarde `false` is, worden ze verborgen. De standaardwaarde is `true`.

### **Voorbeeld**

In het volgende voorbeeld worden alle objecten in de eerste laag onzichtbaar gemaakt:

```
fl.getDocumentDOM().getTimeline().layers[0].visible = false;
```

# Hoofdstuk 26: Object library

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object library staat voor het deelvenster Bibliotheek. Het is een eigenschap van het object Document (zie `document.library`) en is toegankelijk voor `fl.getDocumentDOM().library`.

Het object library bevat een array van items van verschillende typen, zoals symbolen, bitmaps, geluiden en video.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Library:

Methoden	Beschrijving
<code>library.addItemToDocument()</code>	Hiermee wordt het huidige of opgegeven item op de opgegeven positie toegevoegd aan het werkgebied.
<code>library.addNewItem()</code>	Hiermee wordt een nieuw item van het opgegeven type in het deelvenster Bibliotheek gemaakt en wordt het nieuwe item ingesteld op het geselecteerde item.
<code>library.deleteItem()</code>	Hiermee worden de huidige items of wordt een opgegeven item verwijderd uit het deelvenster Bibliotheek.
<code>library.duplicateItem()</code>	Hiermee wordt een kopie gemaakt van het geselecteerde of opgegeven item.
<code>library.editItem()</code>	Hiermee wordt het geselecteerde of opgegeven item geopend in de bewerkingsmodus.
<code>library.expandFolder()</code>	Hiermee wordt de geselecteerde of opgegeven map in de bibliotheek uit- of ingevouwen.
<code>library.findItemIndex()</code>	Hiermee wordt de indexwaarde geretourneerd van het bibliotheekitem (met basis nul).
<code>library.getItemProperty()</code>	Hiermee wordt de eigenschap voor het geselecteerde item opgehaald.
<code>library.getItemType()</code>	Hiermee wordt het type object opgehaald dat is geselecteerd of opgegeven door een bibliotheekpad.
<code>library.getSelectedItems()</code>	Hiermee wordt de array van alle geselecteerde items in de bibliotheek opgehaald.
<code>library.importEmbeddedSWF()</code>	Hiermee wordt een SWF-bestand in de bibliotheek geïmporteerd als gecompileerde clip.
<code>library.itemExists()</code>	Hiermee wordt gecontroleerd of een opgegeven item in de bibliotheek aanwezig is.
<code>library.moveToFolder()</code>	Hiermee wordt het geselecteerde of opgegeven bibliotheekitem verplaatst naar een specifieke map.
<code>library.newFolder()</code>	Hiermee wordt een nieuwe map gemaakt met de opgegeven naam of een standaardnaam ("untitled folder #") als er geen parameter <code>folderName</code> is opgegeven, in de geselecteerde map.
<code>library.renameItem()</code>	Hiermee wordt de naam gewijzigd van het bibliotheekitem dat is geselecteerd in het deelvenster Bibliotheek.

Methode	Beschrijving
<code>library.selectAll()</code>	Hiermee worden alle items in de bibliotheek geselecteerd of wordt de selectie opgeheven.
<code>library.selectItem()</code>	Hiermee wordt een specifiek bibliotheekitem geselecteerd.
<code>library.selectNone()</code>	Hiermee wordt de selectie van alle bibliotheekitems opgeheven.
<code>library.setItemProperty()</code>	Hiermee wordt de eigenschap ingesteld voor alle geselecteerde bibliotheekitems (mappen worden genegeerd).
<code>library.updateItem()</code>	Hiermee wordt het opgegeven item bijgewerkt.

### Overzicht van eigenschappen voor het object library

De volgende eigenschap is beschikbaar voor het object Library:

Eigenschap	Beschrijving
<code>library.items</code>	Een array met objecten Item in de bibliotheek

## library.addItemToDocument()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.addItemToDocument(position [, namePath])
```

### Parameters

**position** Een punt dat de *x,y*-positie van het middelpunt van het item op het werkgebied aangeeft.

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Als *namePath* niet is opgegeven, wordt de huidige bibliotheekselectie gebruikt. Deze parameter is optioneel.

### Geretournerde waarde

Een Booleaanse waarde: `true` als het item aan het document is toegevoegd, anders `false`.

### Beschrijving

Methode; hiermee wordt het huidige of opgegeven item op de opgegeven positie toegevoegd aan het werkgebied.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde item aan het werkgebied toegevoegd op de positie (3, 60):

```
fl.getDocumentDOM().library.addItemToDocument({x:3, y:60});
```

In het volgende voorbeeld wordt het item `Symbol1` in folder1 van de bibliotheek toegevoegd aan het werkgebied op de positie (550, 485):

```
fl.getDocumentDOM().library.addItemToDocument({x:550.0, y:485.0}, "folder1/Symbol1");
```

## library.addNewItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.addNewItem(type [, namePath])
```

### Parameters

**type** Een tekenreeks die het type item aangeeft dat moet worden gemaakt. Voor *type* zijn alleen de volgende waarden acceptabel: "video", "filmclip", "knop", "afbeelding", "bitmap", "scherm" en "map". U kunt dus met deze methode bijvoorbeeld geen geluid aan de bibliotheek toevoegen. Als u een mappad opgeeft, is dat hetzelfde als het gebruiken van [library.newFolder\(\)](#) voordat u deze methode aanroept.

**namePath** Een tekenreeks die de naam van het toe te voegen item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Deze parameter is optioneel.

### Geretoureerde waarde

Een Booleaanse waarde: `true` als het item is gemaakt, anders `false`.

### Beschrijving

Methode; hiermee wordt een nieuw item gemaakt van het opgegeven type in het deelvenster Bibliotheek en wordt het nieuwe item ingesteld op het geselecteerde item. Zie [document.importFile\(\)](#) voor meer informatie over het importeren van items in de bibliotheek, ook geluiden.

### Voorbeeld

In het volgende voorbeeld wordt een nieuwe knop gemaakt met de naam `start` in een nieuwe map met de naam `folderTwo`:

```
fl.getDocumentDOM().library.addNewItem("button", "folderTwo/start");
```

## library.deleteItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.deleteItem([namePath])
```

### Parameters

**namePath** Een tekenreeks die de naam van het te verwijderen item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Als u een mapnaam doorgeeft, worden de map en alle bijbehorende items verwijderd. Als er geen naam is opgegeven, verwijdert Flash het geselecteerde item of de geselecteerde items. Als u alle items in het deelvenster Bibliotheek wilt verwijderen, selecteert u alle items voordat u deze methode gebruikt. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de items zijn verwijderd, anders `false`.

### Beschrijving

Methode; hiermee worden de huidige items of een opgegeven item verwijderd uit het deelvenster Bibliotheek. Deze methode kan betrekking hebben op meerdere items als er meerdere zijn geselecteerd.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde item verwijderd:

```
fl.getDocumentDOM().library.deleteItem();
```

In het volgende voorbeeld wordt het item `Symbol_1` verwijderd uit de bibliotheekmap `Folder_1`:

```
fl.getDocumentDOM().library.deleteItem("Folder_1/Symbol_1");
```

## library.duplicateItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.duplicateItem( [ namePath ] )
```

### Parameters

**namePath** Een tekenreeks die de naam van het te dupliceren item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het item is gedupliceerd, anders `false`. Als er meer dan een item is geselecteerd, retourneert Flash `false`.

### Beschrijving

Methode; hiermee wordt een kopie gemaakt van het geselecteerde of opgegeven item. Het nieuwe item heeft een standaardnaam (bijvoorbeeld `kopie van item`) en het wordt ingesteld als het geselecteerde item. Als er meer dan een item is geselecteerd, mislukt de opdracht.

### Voorbeeld

In het volgende voorbeeld wordt een kopie van het item `square` gemaakt in de bibliotheekmap `test`:

```
fl.getDocumentDOM().library.duplicateItem("test/square");
```

## library.editItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.editItem([namePath])
```

### Parameters

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Als *namePath* niet is opgegeven, wordt het enige geselecteerde bibliotheekitem in de bewerkingsmodus geopend. Als geen of meer dan een item in de bibliotheek is geselecteerd, wordt de eerste scène in de hoofdtijdlijn weergegeven om te worden bewerkt. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het opgegeven item bestaat en kan worden bewerkt, anders `false`.

### Beschrijving

Methode; hiermee wordt het geselecteerde of opgegeven geopend item in de bewerkingsmodus.

### Voorbeeld

In het volgende voorbeeld wordt het item `circle` in de map `test` van de bibliotheek geopend om te worden bewerkt:

```
fl.getDocumentDOM().library.editItem("test/circle");
```

## library.expandFolder()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.expandFolder(bExpand [, bRecurseNestedParents [, namePath]])
```

### Parameters

**bExpand** Een Booleaanse waarde: bij `true` wordt de map uitgevouwen; bij `false` (standaardinstelling) wordt de map samengevouwen.

**bRecurseNestedParents** Een Booleaanse waarde: bij `true` worden alle mappen in de opgegeven map uitgevouwen of samengevouwen, afhankelijk van de waarde van *bExpand*. De standaardwaarde is `false`. Deze parameter is optioneel.

**namePath** Een tekenreeks die de naam en eventueel het pad van de map aangeeft die moet worden uitgevouwen of samengevouwen. Als deze parameter niet is opgegeven, is de methode van toepassing op de geselecteerde map. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het item met succes is samengevouwen of uitgevouwen; `false` als dit is mislukt of als het opgegeven item geen map is.

### Beschrijving

Methode; hiermee wordt de geselecteerde of opgegeven map in de bibliotheek uit- of samengevouwen.

### Voorbeeld

In het volgende voorbeeld wordt de testmap in de bibliotheek samengevouwen, evenals de mappen in de testmap (indien aanwezig):

```
fl.getDocumentDOM().library.expandFolder(false, true, "test");
```

## library.findItemIndex()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.findItemIndex(namePath)
```

### Parameters

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes.

### Geretourneerde waarde

Een geheel getal dat staat voor de indexwaarde (basis nul) van het item.

### Beschrijving

Methode; hiermee wordt de indexwaarde geretourneerd van het bibliotheekitem (met basis nul). De bibliotheekindex is plat, dus mappen worden beschouwd als onderdeel van de hoofdindex. U kunt mappaden gebruiken om een genest item op te geven.

### Voorbeeld

In het volgende voorbeeld wordt de indexwaarde (basis nul) van het bibliotheekitem square opgeslagen (bevindt zich in de testmap) in de variabele `sqIndex`, waarna de indexwaarde in een dialoogvenster wordt weergegeven:

```
var sqIndex = fl.getDocumentDOM().library.findItemIndex("test/square");  
alert(sqIndex);
```

## library.getItemProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.getItemProperty(property)
```

### Parameters

**eigenschap** Een tekenreeks. Een lijst met waarden die u kunt gebruiken als parameter *property*, vindt u in het eigenschappenoverzicht voor het [Object Item](#), samen met de eigenschapoverzichten voor de subklassen ervan.



### Geretourneerde waarde

Een tekenreekswaarde voor de eigenschap.

### Beschrijving

Methode; hiermee wordt de eigenschap voor het geselecteerde item opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt een dialoogvenster weergegeven met de waarde van de koppelings-id voor het symbool wanneer ernaar wordt verwezen met ActionScript of voor delen tijdens runtime:

```
alert(fl.getDocumentDOM().library.getItemProperty("linkageIdentifier"));
```

## library.getItemType()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.getItemType([namePath])
```

### Parameters

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Als *namePath* niet is opgegeven, levert Flash het type van de huidige selectie. Als er meer dan een item is geselecteerd en er geen *namePath* is opgegeven, negeert Flash de opdracht. Deze parameter is optioneel.

### Geretourneerde waarde

Een tekenreekswaarde die het type object aangeeft. Zie [item.itemType](#) voor mogelijke retourwaarden.

### Beschrijving

Methode; hiermee wordt het type object opgehaald dat is geselecteerd of is opgegeven aan de hand van een bibliotheekpad.

### Voorbeeld

In het volgende voorbeeld wordt een dialoogvenster weergegeven met het itemtype `Symbol_1` in de map `Folder_1/Folder_2`:

```
alert(fl.getDocumentDOM().library.getItemType("Folder_1/Folder_2/Symbol_1"));
```

## library.getSelectedItems()

### Beschikbaarheid

Flash MX 2004.

### Parameters

Geen.

### Geretourneerde waarde

Een array van waarden voor alle nu geselecteerde items in de bibliotheek.

### Beschrijving

Methode; hiermee wordt de array van alle geselecteerde items in de bibliotheek opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt de array met geselecteerde bibliotheekitems (in dit geval diverse audiobestanden) opgeslagen in de variabele `selItems` en wordt vervolgens de eigenschap `sampleRate` van het eerste audiobestand in de array gewijzigd in 11 kHz:

```
var selItems = fl.getDocumentDOM().library.getSelectedItems();  
selItems[0].sampleRate = "11 kHz";
```

## library.importEmbeddedSWF()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.importEmbeddedSWF(linkageName, swfData [, libName])
```

### Parameters

**linkageName** Een tekenreeks die de naam levert van de SWF-koppeling van de hoofdfilmclip.

**swfData** Een array van binaire SWF-gegevens, die afkomstig zijn uit een externe bibliotheek of DLL.

**libName** Een tekenreeks die de bibliotheeknaam voor het nieuwe item aangeeft. Als de naam al in gebruik is, maakt de methode een alternatieve naam. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een SWF-bestand in de bibliotheek geïmporteerd als gecompileerde clip. In tegenstelling tot Bestand > Importeren > SWF, kunt u met deze methode een gecompileerd SWF-bestand in de bibliotheek insluiten. Er is geen overeenkomstige functionaliteit voor een gebruikersinterface en u moet deze methode gebruiken met een externe bibliotheek of DLL (zie "[Uitbreidbaarheid in C](#)" op pagina 540).

Het SWF-bestand dat u importeert, moet een filmclip op hoofdniveau met alle inhoud bevatten. Voor die filmclip moet de koppelings-id zijn ingesteld op dezelfde waarde als de parameter *linkageName* die aan deze methode is doorgegeven.

### Voorbeeld

In het volgende voorbeeld wordt het SWF-bestand met de waarde `MyMovie` voor *linkageName* aan de bibliotheek toegevoegd als gecompileerde clip met de naam `Intro`:

```
fl.getDocumentDOM().library.importEmbeddedSWF("MyMovie", swfData, "Intro");
```

## library.itemExists()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.itemExists(namePath)
```

### Parameters

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes.

### Geretoureerde waarde

Een Booleaanse waarde: `true` als het opgegeven item in de bibliotheek aanwezig is, anders `false`.

### Beschrijving

Methode; hiermee wordt gecontroleerd of een opgegeven item in de bibliotheek aanwezig is.

### Voorbeeld

In het volgende voorbeeld wordt `true` weergegeven in een dialoogvenster als het item `Symbol_1` aanwezig is in de bibliotheekmap `Folder_1`, anders `false`:

```
alert(fl.getDocumentDOM().library.itemExists('Folder_1/Symbol_1'));
```

## library.items

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.items
```

### Beschrijving

Eigenschap; een array van itemobjecten in de bibliotheek.

### Voorbeeld

In het volgende voorbeeld wordt de array van alle bibliotheekitems in de variabele `itemArray` opgeslagen:

```
var itemArray = fl.getDocumentDOM().library.items;
```

## library.moveToFolder()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.moveToFolder(folderPath [, itemToMove [, bReplace]])
```

### Parameters

**folderPath** Een tekenreeks die het pad naar de map aangeeft in de vorm "Mapnaam" of "Mapnaam/Mapnaam". Als u een item naar het hoofdniveau wilt verplaatsen, geeft u een lege tekenreeks op (""), voor *folderPath*.

**itemToMove** Een tekenreeks die de naam van het te verplaatsen item aangeeft. Als *itemToMove* niet is opgegeven, worden de nu geselecteerde items verplaatst. Deze parameter is optioneel.

**bReplace** Een Booleaanse waarde. Als er al een item met dezelfde naam bestaat, wordt het bestaande item vervangen door het te verplaatsen item als u *true* opgeeft voor de parameter *bReplace*. Bij *false* wordt de naam van het verwijderde gewijzigd in een unieke naam. De standaardwaarde is *false*. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: *true* als het item is verplaatst, anders *false*.

### Beschrijving

Methode; hiermee wordt het geselecteerde of opgegeven bibliotheekitem verplaatst naar een specifieke map. Als de parameter *folderPath* leeg is, worden de items naar het hoofdniveau verplaatst.

### Voorbeeld

In het volgende voorbeeld wordt het item `Symbol_1` naar de bibliotheekmap `new` verplaatst en wordt het item met dezelfde naam in die map vervangen:

```
fl.getDocumentDOM().library.moveToFolder("new", "Symbol_1", true);
```

## library.newFolder()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.newFolder([folderPath])
```

### Parameters

**folderPath** Een tekenreeks die de naam van de te maken map aangeeft. Als een pad is opgegeven en het pad niet bestaat, wordt dit gemaakt. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: *true* als het item is gemaakt, anders *false*.

### Beschrijving

Methode; hiermee wordt een nieuwe map gemaakt met de opgegeven naam of een standaardnaam ("untitled folder #") als er geen parameter *folderName* is opgegeven, in de geselecteerde map.

### Voorbeeld

In het volgende voorbeeld worden twee nieuwe bibliotheekmappen gemaakt. De tweede map is een submap van de eerste:

```
fl.getDocumentDOM().library.newFolder("first/second");
```

## library.renameItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.renameItem(name)
```

### Parameters

**name** Een tekenreeks die een nieuwe naam voor het bibliotheekitem aangeeft.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de naam van het item is gewijzigd, anders `false`. Als er meerdere items zijn geselecteerd, worden er geen namen gewijzigd en is de geretourneerde waarde `false` (overeenkomstig het gedrag van de gebruikersinterface).

### Beschrijving

Methode; hiermee wordt de naam gewijzigd van het bibliotheekitem dat is geselecteerd in het deelvenster Bibliotheek.

### Voorbeeld

In het volgende voorbeeld krijgt het geselecteerde bibliotheekitem de naam `new name`:

```
fl.getDocumentDOM().library.renameItem("new name");
```

## library.selectAll()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.selectAll([bSelectAll])
```

### Parameters

**bSelectAll** Een Booleaanse waarde die aangeeft of alle items in de bibliotheek moeten worden geselecteerd of gedeselecteerd. Laat deze parameter weg of gebruik de standaardwaarde `true` om alle items in de bibliotheek te selecteren; met `false` worden alle bibliotheekitems gedeselecteerd. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle items in de bibliotheek geselecteerd of gedeselecteerd.

### Voorbeeld

In het volgende voorbeeld worden alle items in de bibliotheek geselecteerd:

```
fl.getDocumentDOM().library.selectAll();  
fl.getDocumentDOM().library.selectAll(true);
```

In de volgende voorbeelden worden alle items in de bibliotheek gedeselecteerd:

```
fl.getDocumentDOM().library.selectAll(false);  
fl.getDocumentDOM().library.selectNone();
```

## library.selectItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.selectItem(namePath [, bReplaceCurrentSelection [, bSelect]])
```

### Parameters

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes.

**bReplaceCurrentSelection** Een Booleaanse waarde die aangeeft of de huidige selectie moet worden vervangen of dat het item aan de huidige selectie moet worden toegevoegd. De standaardwaarde is `true` (huidige selectie vervangen). Deze parameter is optioneel.

**bSelect** Een Booleaanse waarde die aangeeft of een item moet worden geselecteerd of gedeselecteerd. De standaardwaarde is `true` (selecteren). Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het opgegeven item aanwezig is, anders `false`.

### Beschrijving

Methode; hiermee wordt een specifiek bibliotheekitem geselecteerd.

### Voorbeeld

In het volgende voorbeeld wordt de huidige selectie in de bibliotheek gewijzigd in `Symbol_1` in `untitled Folder_1`:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1");
```

In het volgende voorbeeld wordt de huidige selectie in de bibliotheek uitgebreid met `Symbol_1` in `untitled Folder_1`:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", false);
```

In het volgende voorbeeld wordt de selectie van `Symbol_1` in `untitled Folder_1` ongedaan gemaakt en worden andere geselecteerde items niet gewijzigd:

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", true, false);
```

## library.selectNone()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.selectNone()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle bibliotheekitems gedeselecteerd.

### Voorbeeld

In de volgende voorbeelden worden alle items in de bibliotheek gedeselecteerd:

```
fl.getDocumentDOM().library.selectNone();  
fl.getDocumentDOM().library.selectAll(false);
```

## library.setItemProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.setItemProperty(property, value)
```

### Parameters

**eigenschap** Een tekenreeks die de naam is van de eigenschap die moet worden ingesteld. Een lijst met eigenschappen vindt u in het eigenschappenoverzicht voor het [Object Item](#) en in de eigenschappenoverzichten voor de bijbehorende subklassen. Zie "[Overzicht van de DOM-structuur](#)" op pagina 11 als u wilt zien welke objecten subklassen van het object Item zijn.

**value** De waarde die aan de opgegeven eigenschap moet worden toegewezen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de eigenschap ingesteld voor alle geselecteerde bibliotheekitems (mappen worden genegeerd).

### Voorbeeld

In het volgende voorbeeld wordt de waardeknop toegewezen aan de eigenschap `symbolType` voor het geselecteerde bibliotheekitem of de geselecteerde bibliotheekitems. In dit geval moet het item een [Object SymbolItem](#) zijn; `symbolType` is een geldige eigenschap voor objecten `SymbolItem`.

```
fl.getDocumentDOM().library.setItemProperty("symbolType", "button");
```

## library.updateItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
library.updateItem([namePath])
```

### Parameters

**namePath** Een tekenreeks die de naam van het item aangeeft. Als het item zich in een map bevindt, kunt u de naam en het pad opgeven met slashes. Dit is hetzelfde als wanneer u met de rechtermuisknop op een item klikt en in de gebruikersinterface Bijwerken selecteert in het menu. Als er geen naam is opgegeven, wordt de huidige selectie bijgewerkt. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als Flash het item heeft bijgewerkt, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven item bijgewerkt.

### Voorbeeld

In het volgende voorbeeld wordt een dialoogvenster weergegeven waarin staat of het geselecteerde item is bijgewerkt (`true`) of niet (`false`):

```
alert(fl.getDocumentDOM().library.updateItem());
```



# Hoofdstuk 27: Object Math

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Math is beschikbaar als een alleen-lezen eigenschap van het object flash; zie [fl.Math](#). Dit object levert methoden om algemene wiskundige bewerkingen uit te voeren.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Math:

Methode	Beschrijving
<a href="#">Math.concatMatrix()</a>	Hiermee wordt een matrix samengevoegd en het resultaat geretourneerd.
<a href="#">Math.invertMatrix()</a>	Hiermee wordt het omgekeerde van de opgegeven matrix geretourneerd.
<a href="#">Math.pointDistance()</a>	Hiermee wordt de afstand tussen twee punten berekend.

## Math.concatMatrix()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
Math.concatMatrix(mat1, mat2)
```

### Parameters

**mat1**, **mat2** Geef de objecten Matrix op die moeten worden samengevoegd (zie [Object Matrix](#)). Elke parameter moet een object zijn met de velden a, b, c, d, tx en ty.

### Geretourneerde waarde

Een samengevoegde objectmatrix.

### Beschrijving

Methode; hiermee wordt een matrix samengevoegd en het resultaat geretourneerd.

### Voorbeeld

In het volgende voorbeeld wordt het momenteel geselecteerde object opgeslagen in de variabele `elt`, de objectmatrix vermenigvuldigd met de weergavematrix en die waarde opgeslagen in de variabele `mat`:

```
var elt = fl.getDocumentDOM().selection[0];
var mat = fl.Math.concatMatrix( elt.matrix , fl.getDocumentDOM().viewMatrix );
```

## Math.invertMatrix()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
Math.invertMatrix(mat)
```

### Parameters

**mat** Hiermee worden de objecten Matrix aangegeven die moeten worden omgekeerd (zie [Object Matrix](#)). De parameter moet de volgende velden hebben: a, b, c, d, tx en ty.

### Geretourneerde waarde

Een object Matrix dat het omgekeerde is van de oorspronkelijke matrix.

### Beschrijving

Methode; hiermee wordt het omgekeerde geretourneerd van de opgegeven matrix.

### Voorbeeld

In het volgende voorbeeld wordt het momenteel geselecteerde object opgeslagen in de variabele `elt`, wordt die matrix toegewezen aan de variabele `mat` en het omgekeerde van de matrix opgeslagen in de variabele `inv`:

```
var elt = fl.getDocumentDOM().selection[0];  
var mat = elt.matrix;  
var inv = fl.Math.invertMatrix( mat );
```

## Math.pointDistance()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
Math.pointDistance(pt1, pt2)
```

### Parameters

**pt1, pt2** Geef de punten op waartussen de afstand wordt gemeten.

### Geretourneerde waarde

Een drijvende-kommawaarde die de afstand tussen de punten vertegenwoordigt.

### Beschrijving

Methode; hiermee wordt de afstand tussen twee punten berekend.

### Voorbeeld

In het volgende voorbeeld wordt de waarde voor de afstand tussen *pt1* en *pt2* opgeslagen in de variabele *dist*:

```
var pt1 = {x:10, y:20}  
var pt2 = {x:100, y:200}  
var dist = fl.Math.pointDistance(pt1, pt2);
```

# Hoofdstuk 28: Object Matrix

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Matrix vertegenwoordigt een transformatiematrix.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Matrix:

Eigenschap	Beschrijving
<code>matrix.a</code>	Een drijvende-kommawaarde die het element (0,0) in de transformatiematrix opgeeft.
<code>matrix.b</code>	Een drijvende-kommawaarde die het element (0,1) in de matrix opgeeft.
<code>matrix.c</code>	Een drijvende-kommawaarde die het element (1,0) in de matrix opgeeft.
<code>matrix.d</code>	Een drijvende-kommawaarde die het element (1,1) in de matrix opgeeft.
<code>matrix.tx</code>	Een drijvende-kommawaarde die de locatie op de x-as van het registratiepunt van een symbool of het midden van een vorm opgeeft.
<code>matrix.ty</code>	Een drijvende-kommawaarde die de locatie op de y-as van het registratiepunt van een symbool of het midden van een vorm opgeeft.

## matrix.a

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`matrix.a`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die het element (0,0) in de transformatiematrix opgeeft. Deze waarde vertegenwoordigt de schalingsfactor van de x-as van het object.

### Voorbeeld

De eigenschappen `a` en `d` in een matrix vertegenwoordigen schaling. In het volgende voorbeeld worden de waarden ingesteld op respectievelijk 2 en 3, waardoor het geselecteerde object wordt geschaald naar twee keer de breedte en drie keer de hoogte:

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 2;
mat.d = 3;
fl.getDocumentDOM().selection[0].matrix = mat;
```

U kunt een object draaien door de eigenschappen `a`, `b`, `c` en `d` van de matrix ten opzichte van elkaar in te stellen, waarbij  $a = d$  en  $b = -c$ . De waarden 0.5, 0.8, -0.8 en 0.5 draaien het object bijvoorbeeld 60°:

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.a = 0.5;  
mat.b = 0.8;  
mat.c = 0.8*(-1);  
mat.d = 0.5;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

U kunt instellen dat  $a = d = 1$  en  $c = b = 0$  om de oorspronkelijke vorm van het object te herstellen.

## matrix.b

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`matrix.b`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die het element (0,1) in de matrix opgeeft. Deze waarde vertegenwoordigt de verticale scheeftrekking van een vorm; hierdoor beweegt Flash de rechterrand van de vorm langs de verticale as.

De eigenschappen `matrix.b` en `matrix.c` in een matrix vertegenwoordigen het scheeftrekken (zie [matrix.c](#)).

### Voorbeeld

In het volgende voorbeeld kunt u `b` en `c` instellen op respectievelijk -1 en 0; met deze instellingen wordt het object in een verticale hoek van 45° scheefgetrokken:

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.b = -1;  
mat.c = 0;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

U kunt `b` en `c` instellen op 0 om het object terug te trekken naar de oorspronkelijke vorm.

Zie ook het voorbeeld voor [matrix.a](#).

## matrix.c

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`matrix.c`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die het element (1,0) in de matrix opgeeft. Door deze waarde trekt Flash het object scheef door de onderrand te verschuiven langs een horizontale as.

De eigenschappen `matrix.b` en `matrix.c` in een matrix vertegenwoordigen het scheeffrekken.

### Voorbeeld

Zie het voorbeeld voor [matrix.b](#).

## matrix.d

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`matrix.d`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die het element (1,1) in de matrix opgeeft. Deze waarde vertegenwoordigt de schalingsfactor van de *y*-as van het object.

### Voorbeeld

Zie het voorbeeld voor [matrix.a](#).

## matrix.tx

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`matrix.tx`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de locatie op de *x*-as van het registratiepunt van een symbool (ook wel *oorsprongpunt* of *nulpunt*) of het midden van een vorm opgeeft. Hiermee wordt de *x*-vertaling van de transformatie gedefinieerd.

U kunt een object verplaatsen door de eigenschappen `matrix.tx` en `matrix.ty` in te stellen (zie [matrix.ty](#)).

### Voorbeeld

In het volgende voorbeeld worden `tx` en `ty` ingesteld op 0 waardoor het registratiepunt van het object wordt verplaatst naar punt 0,0 in het document:

```
var mat = fl.getDocumentDOM().selection[0].matrix;  
mat.tx = 0;  
mat.ty = 0;  
fl.getDocumentDOM().selection[0].matrix = mat;
```

## matrix.ty

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`matrix.ty`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de locatie op de  $y$ -as van het registratiepunt van een symbool of het midden van een vorm opgeeft. Hiermee wordt de  $y$ -vertaling van de transformatie gedefinieerd.

U kunt een object verplaatsen door de eigenschappen `matrix.tx` en `matrix.ty` in te stellen.

### Voorbeeld

Zie het voorbeeld voor [matrix.tx](#).

# Hoofdstuk 29: Object outputPanel

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Dit object vertegenwoordigt het deelvenster Uitvoer waarin informatie over probleemoplossingen wordt weergegeven, bijvoorbeeld syntaxisfouten. Gebruik `fl.outputPanel` (of `flash.outputPanel`) om dit object te benaderen. Zie [fl.outputPanel](#).

## Overzicht van methoden

Het object outputPanel gebruikt de volgende methoden:

Methoden	Beschrijving
<code>outputPanel.clear()</code>	Hiermee wordt de inhoud van het deelvenster Uitvoer gewist.
<code>outputPanel.save()</code>	Hiermee wordt de inhoud van het deelvenster Uitvoer in een lokaal tekstbestand opgeslagen.
<code>outputPanel.trace()</code>	Hiermee wordt een regel toegevoegd aan het deelvenster Uitvoer. Deze regel wordt afgesloten met een nieuwe regel.

## outputPanel.clear()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
outputPanel.clear()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methoden; hiermee wordt de inhoud van het deelvenster Uitvoer gewist. U kunt deze methode gebruiken in een toepassing voor batchverwerking om een lijst met fouten te wissen, of u kunt deze methode gebruiken om fouten incrementeel op te slaan met [outputPanel.save\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt de huidige inhoud van het deelvenster Uitvoer gewist:

```
fl.outputPanel.clear();
```



## `outputPanel.save()`

### Beschikbaarheid

Flash MX 2004; parameter `bUseSystemEncoding` toegevoegd in Flash 8.

### Gebruik

```
outputPanel.save(fileURI [, bAppendToFile [ , bUseSystemEncoding]])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het lokale bestand opgeeft dat de inhoud van het deelvenster Uitvoer bevat.

**bAppendToFile** Een optionele Booleaanse waarde. Als de waarde `true` is, wordt de inhoud van het deelvenster Uitvoer toegevoegd aan het uitvoerbestand; als de waarde `false` is, overschrijft de methode het uitvoerbestand als dit al bestaat. De standaardwaarde is `false`.

**bUseSystemEncoding** Een optionele Booleaanse waarde. Als de waarde `true` is, wordt systeemcodering gebruikt om de tekst van het deelvenster Uitvoer op te slaan; als de waarde `false` is, wordt UTF-8-codering gebruikt om de tekst van het deelvenster Uitvoer op te slaan, hierbij staan bytevolgordetekens aan het begin van de tekst. De standaardwaarde is `false`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de inhoud van het deelvenster Uitvoer opgeslagen in een lokaal tekstbestand. Het bestand wordt hierbij overschreven of de tekst wordt aan het bestand toegevoegd.

Als `fileURI` ongeldig is of niet is opgegeven, wordt een fout gerapporteerd.

Deze methode is nuttig voor batchverwerking. U kunt bijvoorbeeld een JSFL-bestand maken waarmee verschillende componenten worden gecompileerd. Eventuele compileerfouten verschijnen in het deelvenster Uitvoer. U kunt met deze methode de resulterende fouten opslaan in een tekstbestand dat automatisch kan worden geparseerd door het gebruikte bouwsysteem.

### Voorbeeld

In het volgende voorbeeld wordt de inhoud van het deelvenster Uitvoer opgeslagen in het bestand `batch.log` file in de map `/tests`. Als er al een `batch.log` bestaat, wordt dit overschreven.

```
fl.outputPanel.save("file:///c:/tests/batch.log");
```

## `outputPanel.trace()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
outputPanel.trace(message)
```

### Parameters

**message** Een tekenreeks die de tekst bevat die moet worden toegevoegd aan het deelvenster Uitvoer.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een teksttekenreeks verzonden naar het deelvenster Uitvoer, afgesloten door een nieuwe regel. Het deelvenster Uitvoer verschijnt als dit nog niet zichtbaar is. Deze methode is gelijk aan `fl.trace()` en werkt op dezelfde wijze als de instructie `trace()` in ActionScript.

Als u een lege regel wilt verzenden, gebruikt u `outputPanel.trace("")` of `outputPanel.trace("\n")`. U kunt de tweede opdracht inline gebruiken door `\n` op te nemen in de tekenreeks *message*.

### Voorbeeld

In het volgende voorbeeld worden meerdere tekstregels in het deelvenster Uitvoer weergegeven:

```
fl.outputPanel.clear();
fl.outputPanel.trace("Hello World!!!");
var myPet = "cat";
fl.outputPanel.trace("\nI have a " + myPet);
fl.outputPanel.trace("");
fl.outputPanel.trace("I love my " + myPet);
fl.outputPanel.trace("Do you have a " + myPet + "?");
```

# Hoofdstuk 30: Object Oval

**Overerving** [Object Element](#) > [Object Shape](#) > Object Oval

## Beschikbaarheid

Flash CS3 Professional.

## Beschrijving

Het object Oval is een vorm die wordt getekend met het gereedschap Ovaal primitief. Gebruik `shape.isOvalObject` om te bepalen of een item een object Oval is.

## Overzicht van eigenschappen

Naast de eigenschappen van het [Object Shape](#) kunt u de volgende eigenschappen gebruiken voor het object Oval. Gebruik `document.setOvalObjectProperty()` om de eigenschappen in te stellen van een object Oval.

Eigenschap	Beschrijving
<code>OvalObject.closePath</code>	Alleen-lezen; een Booleaanse waarde die opgeeft of het selectievakje Pad sluiten in de Eigenschapcontrole is ingeschakeld.
<code>OvalObject.endAngle</code>	Alleen-lezen; een drijvende-kommawaarde die de eindhoek van het object Oval aangeeft.
<code>OvalObject.innerRadius</code>	Alleen-lezen; een drijvende-kommawaarde die de binnenstraal van het object Oval aangeeft als percentage.
<code>OvalObject.startAngle</code>	Alleen-lezen; een drijvende-kommawaarde die de beginhoek van het object Oval aangeeft.

## OvalObject.closePath

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`OvalObject.closePath`

### Beschrijving

Alleen-lezen eigenschap; een Booleaanse waarde die opgeeft of het selectievakje Pad sluiten in de Eigenschapcontrole is ingeschakeld. Als de beginhoek en eindhoek voor het object gelijk zijn, heeft het instellen van deze eigenschap geen effect tot de waarden veranderen.

Gebruik `document.setOvalObjectProperty()` om deze waarde in te stellen.

### Voorbeeld

In het volgende voorbeeld wordt de selectie van de eigenschap `OvalObject.closePath` ongedaan gemaakt:

```
fl.getDocumentDOM().setOvalObjectProperty("closePath", false);
```

### Zie ook

[document.setOvalObjectProperty\(\)](#), [shape.isOvalObject](#)

## OvalObject.endAngle

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`OvalObject.endAngle`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de eindhoek van het object Oval aangeeft. Acceptabele waarden lopen van 0 tot en met 360.

Gebruik [document.setOvalObjectProperty\(\)](#) om deze waarde in te stellen.

### Voorbeeld

In het volgende voorbeeld wordt de eindhoek van geselecteerde objecten Oval ingesteld op 270.

```
fl.getDocumentDOM().setOvalObjectProperty("endAngle", 270);
```

### Zie ook

[document.setOvalObjectProperty\(\)](#), [OvalObject.startAngle](#), [shape.isOvalObject](#)

## OvalObject.innerRadius

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`OvalObject.innerRadius`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de binnenstraal van het object Oval aangeeft als percentage. Acceptabele waarden lopen van 0 tot en met 99.

Gebruik [document.setOvalObjectProperty\(\)](#) om deze waarde in te stellen.

### Voorbeeld

In het volgende voorbeeld wordt de binnenstraal van geselecteerde objecten Oval ingesteld op 50 procent:

```
fl.getDocumentDOM().setOvalObjectProperty("innerRadius", 50);
```

### Zie ook

[document.setOvalObjectProperty\(\)](#), [shape.isOvalObject](#)

## OvalObject.startAngle

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`OvalObject.startAngle`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de beginhoek van het object Oval aangeeft. Acceptabele waarden lopen van 0 tot en met 360.

Gebruik `document.setOvalObjectProperty()` om deze waarde in te stellen.

### Voorbeeld

In het volgende voorbeeld wordt de beginhoek van geselecteerde objecten Oval ingesteld op 270:

```
fl.getDocumentDOM().setOvalObjectProperty("startAngle", 270);
```

### Zie ook

[document.setOvalObjectProperty\(\)](#), [OvalObject.endAngle](#), [shape.isOvalObject](#)

# Hoofdstuk 31: Object Parameter

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het objecttype Parameter wordt geopend vanuit de array `componentInstance.parameters` (die overeenkomt met de Eigenschapcontrole voor componenten in het ontwerpprogramma).

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Parameter:

Methode	Beschrijving
<code>parameter.insertItem()</code>	Hiermee wordt een item ingevoegd in een lijst, object of array.
<code>parameter.removeItem()</code>	Hiermee wordt een element verwijderd uit het lijst-, object- of arraytype van een scherm- of componentparameter.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Parameter:

Eigenschap	Beschrijving
<code>parameter.category</code>	Een tekenreeks die de eigenschap <code>category</code> opgeeft voor de parameter <code>screen</code> of de parameter <code>componentInstance</code> .
<code>parameter.listIndex</code>	Een geheel getal dat de waarde van het geselecteerde item in de lijst opgeeft.
<code>parameter.name</code>	Alleen-lezen; een tekenreeks die de naam van de parameter opgeeft.
<code>parameter.value</code>	Komt overeen met het veld Waarde op het tabblad Parameters van de Componentcontrole, het tabblad Parameters van de Eigenschapcontrole of het scherm Eigenschapcontrole.
<code>parameter.valueType</code>	Alleen-lezen; een tekenreeks die het type van de scherm- of componentparameter opgeeft.
<code>parameter.verbose</code>	Hiermee wordt opgegeven of de parameter wordt weergegeven.

## parameter.category

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`parameter.category`

### Beschrijving

Eigenschap; een tekenreeks die de eigenschap `category` opgeeft voor de parameter `screen` of de parameter `componentInstance`. Deze eigenschap biedt een andere manier om een lijst met parameters te presenteren. Deze functionaliteit is niet beschikbaar via de gebruikersinterface van Flash.

## parameter.insertItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
parameter.insertItem(index, name, value, type)
```

### Parameters

**index** Een op nul gebaseerde index met gehele getallen die aangeeft waar het item wordt ingevoegd in de lijst, het object of de array. Als de index 0 is, wordt het item ingevoegd aan het begin van de lijst. Als de index groter is dan de lengte van de lijst, wordt het nieuwe item ingevoegd aan het einde van de lijst.

**name** Een tekenreeks die de naam opgeeft van het item dat moet worden ingevoegd. Dit is een vereiste parameter voor objectparameters.

**value** Een tekenreeks die de waarde opgeeft van het item dat moet worden ingevoegd.

**type** Een tekenreeks die het type opgeeft van het item dat moet worden ingevoegd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een item ingevoegd in een lijst, object of array. Als een parameter een lijst, object of array is, is de eigenschap *value* een array.

### Voorbeeld

In het volgende voorbeeld wordt de waarde `New Value` ingevoegd in de parameter `labelPlacement`:

```
// Select an instance of a Button component on the Stage.
var parms = fl.getDocumentDOM().selection[0].parameters;
parms[2].insertItem(0, "name", "New Value", "String");
var values = parms[2].value;
for(var prop in values){
    fl.trace("labelPlacement parameter value = " + values[prop].value);
}
```

## parameter.listIndex

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
parameter.listIndex
```

### Beschrijving

Eigenschap; de waarde van het geselecteerde item in de lijst. Deze eigenschap is alleen geldig als `parameter.valueType"List"` is.

### Voorbeeld

In het volgende voorbeeld wordt de eerste parameter voor een dia ingesteld, dit is de parameter `autoKeyNav`. Om de parameter in te stellen op een van de acceptabele waarden (`true`, `false` of `inherit`), is `parameter.listIndex` ingesteld op de index van het item in de lijst (0 voor `true`, 1 voor `false`, 2 voor `inherit`).

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;  
parms[0].listIndex = 1;
```

## parameter.name

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
parameter.name
```

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die de naam van de parameter opgeeft.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de vijfde parameter getoond voor de geselecteerde component:

```
var parms = fl.getDocumentDOM().selection[0].parameters;  
fl.trace("name: " + parms[4].name);
```

In het volgende voorbeeld wordt de naam van de vijfde parameter getoond voor het opgegeven scherm:

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters; fl.trace("name: " +  
parms[4].name);
```

## parameter.removeItem()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
parameter.removeItem(index)
```

### Parameters

**index** De op nul gebaseerde index met gehele getallen van het item dat u wilt verwijderen uit de scherm- of componenteigenschap.

### Geretourneerde waarde

Niets.



### Beschrijving

Methode; hiermee wordt een element verwijderd uit het lijst-, object- of arraytype van een scherm- of componentparameter.

### Voorbeeld

In het volgende voorbeeld wordt het element op index 1 verwijderd uit de parameter `labelPlacement` van een component:

```
// Select an instance of a Button component on the Stage.
var parms = fl.getDocumentDOM().selection[0].parameters;
var values = parms[2].value;
fl.trace("--Original--");
for(var prop in values){
fl.trace("labelPlacement value = " + values[prop].value);
}
parms[2].removeItem(1);

var newValues = parms[2].value;
fl.trace("--After Removing Item--");
for(var prop in newValues){
fl.trace("labelPlacement value = " + newValues[prop].value);
}
```

In het volgende voorbeeld wordt het element op index 1 verwijderd uit de parameter `autoKeyNav` van een scherm:

```
// Open a presentation document.
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;
var values = parms[0].value;
fl.trace("--Original--");
for(var prop in values){
fl.trace("autoKeyNav value = " + values[prop].value);
}
parms[0].removeItem(1);

var newValues = parms[0].value;
fl.trace("--After Removing Item--");
for(var prop in newValues){
fl.trace("autoKeyNav value = " + newValues[prop].value);
}
```

## parameter.value

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`parameter.value`

### Beschrijving

Eigenschap; komt overeen met het veld Waarde op het tabblad Parameters van de Componentcontrole, het tabblad Parameters van de Eigenschapcontrole of het scherm Eigenschapcontrole. Het type van de eigenschap `value` wordt bepaald door de eigenschap `valueType` voor de parameter (zie [parameter.valueType](#)).

## parameter.valueType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`parameter.valueType`

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het type van de parameter van het scherm of de component opgeeft. Het type kan een van de volgende waarden hebben: "Default", "Array", "Object", "List", "String", "Number", "Boolean", "Font Name", "Color", "Collection", "Web Service URL" of "Web Service Operation".

### Zie ook

[parameter.value](#)

## parameter.verbose

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`parameter.verbose`

### Beschrijving

Eigenschap; hiermee wordt opgegeven of de parameter wordt weergegeven. Als de waarde van deze eigenschap 0 (nonverbose) is, wordt de parameter alleen weergegeven in de Componentcontrole. Als de waarde 1 (verbose) is, wordt de parameter weergegeven in de Componentcontrole en op het tabblad Parameters van de Eigenschapcontrole.

# Hoofdstuk 32: Object Path

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Path definieert een reeks lijnsegmenten (recht, gekromd of beide) die u in het algemeen gebruikt wanneer u uitbreidbare gereedschappen maakt. In het volgende voorbeeld wordt een instantie getoond van een object Path dat wordt geretourneerd door het object flash:

```
path = fl.drawingLayer.newPath();
```

Zie ook [Object drawingLayer](#).

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Path:

Methode	Beschrijving
<code>path.addCubicCurve()</code>	Hiermee wordt een cubisch Bézier-curve segment toegevoegd aan het pad.
<code>path.addCurve()</code>	Hiermee wordt een kwadratisch Bézier-curve segment toegevoegd aan het pad.
<code>path.addPoint()</code>	Hiermee wordt een punt toegevoegd aan het pad.
<code>path.clear()</code>	Hiermee worden alle punten uit het pad verwijderd.
<code>path.close()</code>	Hiermee wordt een punt toegevoegd op de locatie van het eerste punt van het pad en wordt het pad uitgebreid naar dat punt waardoor het pad wordt gesloten.
<code>path.makeShape()</code>	Hiermee wordt een vorm gemaakt op het werkgebied met de huidige streek- en vulinstellingen.
<code>path.newContour()</code>	Hiermee wordt een nieuwe contour in het pad gestart.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Path:

Eigenschap	Beschrijving
<code>path.nPts</code>	Alleen-lezen; hiermee wordt een geheel getal geretourneerd dat het aantal punten in het pad vertegenwoordigt.

## path.addCubicCurve()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.addCubicCurve(xAnchor, yAnchor, x2, y2, x3, y3, x4, y4)
```

### Parameters

**xAnchor** Een drijvende-kommawaarde die de *x*-positie opgeeft van het eerste besturingspunt.

**yAnchor** Een drijvende-kommawaarde die de *y*-positie opgeeft van het eerste besturingspunt.

**x2** Een drijvende-kommawaarde die de *x*-positie opgeeft van het tweede besturingspunt.

**y2** Een drijvende-kommawaarde die de *y*-positie opgeeft van het tweede besturingspunt.

**x3** Een drijvende-kommawaarde die de *x*-positie opgeeft van het derde besturingspunt.

**y3** Een drijvende-kommawaarde die de *y*-positie opgeeft van het derde besturingspunt.

**x4** Een drijvende-kommawaarde die de *x*-positie opgeeft van het vierde besturingspunt.

**y4** Een drijvende-kommawaarde die de *y*-positie opgeeft van het vierde besturingspunt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een cubisch Bézier-curvesegment toegevoegd aan het pad.

### Voorbeeld

In het volgende voorbeeld wordt een nieuw pad gemaakt en opgeslagen in de variabele `myPath`, en wordt de curve toegewezen aan het pad:

```
var myPath = fl.drawingLayer.newPath();  
myPath.addCubicCurve(0, 0, 10, 20, 20, 20, 30, 0);
```

## path.addCurve()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.addCurve(xAnchor, yAnchor, x2, y2, x3, y3)
```

### Parameters

**xAnchor** Een drijvende-kommawaarde die de *x*-positie opgeeft van het eerste besturingspunt.

**yAnchor** Een drijvende-kommawaarde die de *y*-positie opgeeft van het eerste besturingspunt.

**x2** Een drijvende-kommawaarde die de *x*-positie opgeeft van het tweede besturingspunt.

**y2** Een drijvende-kommawaarde die de *y*-positie opgeeft van het tweede besturingspunt.

**x3** Een drijvende-kommawaarde die de *x*-positie opgeeft van het derde besturingspunt.

**y3** Een drijvende-kommawaarde die de *y*-positie opgeeft van het derde besturingspunt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een kwadratisch Bézier-curvesegment toegevoegd aan het pad.

### Voorbeeld

In het volgende voorbeeld wordt een nieuw pad gemaakt en opgeslagen in de variabele `myPath`, en wordt de curve toegewezen aan het pad:

```
var myPath = fl.drawingLayer.newPath();  
myPath.addCurve(0, 0, 10, 20, 20, 0);
```

## path.addPoint()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.addPoint(x, y)
```

### Parameters

**x** Een drijvende-kommawaarde die de *x*-positie opgeeft van het punt.

**y** Een drijvende-kommawaarde die de *y*-positie opgeeft van het punt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een punt toegevoegd aan het pad.

### Voorbeeld

In het volgende voorbeeld wordt een nieuw pad gemaakt en opgeslagen in de variabele `myPath`, en wordt het nieuwe punt toegewezen aan het pad:

```
var myPath = fl.drawingLayer.newPath();  
myPath.addPoint(10, 100);
```

## path.clear()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.clear()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle punten uit het pad verwijderd.

### Voorbeeld

In het volgende voorbeeld worden alle punten verwijderd uit een pad dat is opgeslagen in de variabele `myPath`:

```
var myPath = fl.drawingLayer.newPath();  
myPath.clear();
```

## path.close()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.close()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een punt toegevoegd op de locatie van het eerste punt van het pad en wordt het pad uitgebreid naar dat punt waardoor het pad wordt gesloten. Als het pad geen punten heeft, worden geen punten toegevoegd.

### Voorbeeld

In het volgende voorbeeld wordt een gesloten pad gemaakt:

```
var myPath = fl.drawingLayer.newPath();  
myPath.close();
```

## path.makeShape()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.makeShape([bSupressFill [, bSupressStroke]])
```

### Parameters

**bSuppressFill** Een Booleaanse waarde. Als de waarde is ingesteld op `true`, wordt de vulling onderdrukt die zou worden toegepast op de vorm. De standaardwaarde is `false`. Deze parameter is optioneel.

**bSuppressStroke** Een Booleaanse waarde. Als de waarde is ingesteld op `true`, wordt de streek onderdrukt die zou worden toegepast op de vorm. De standaardwaarde is `false`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een vorm gemaakt op het werkgebied met de huidige streek- en vulinstellingen. Het pad wordt gewist nadat de vorm is gemaakt. Deze methode heeft twee optionele parameters voor het onderdrukken van de vulling en streek voor het resulterende vormobject. Als u deze parameters weglaat of instelt op `false`, wordt de huidige waarde voor de vulling en streek gebruikt.

### Voorbeeld

In de volgende voorbeelden wordt een vorm gemaakt met de huidige vulling en geen streek:

```
var myPath = fl.drawingLayer.newPath();  
myPath.makeShape(false, true);
```

## path.newContour()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
path.newContour()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een nieuwe contour in het pad gestart.

### Voorbeeld

In het volgende voorbeeld wordt een hol vierkant gemaakt:

```
var myPath = fl.drawingLayer.newPath();
myPath.addPoint(0, 0);
myPath.addPoint(0, 30);
myPath.addPoint(30, 30);
myPath.addPoint(30, 0);
myPath.addPoint(0, 0);

myPath.newContour();
myPath.addPoint(10, 10);
myPath.addPoint(10, 20);
myPath.addPoint(20, 20);
myPath.addPoint(20, 10);
myPath.addPoint(10, 10);

myPath.makeShape();
```

## path.nPts

### Beschikbaarheid

Flash MX 2004.

### Gebruik

path.nPts

### Beschrijving

Alleen-lezen eigenschap; hiermee wordt een geheel getal geretourneerd dat het aantal punten in het pad vertegenwoordigt. Een nieuw pad heeft 0 punten.

### Voorbeeld

In het volgende voorbeeld wordt in het deelvenster Uitvoer het aantal punten weergegeven in het pad waarnaar wordt verwezen door de variabele `myPath`:

```
var myPath = fl.drawingLayer.newPath();
var numOfPoints = myPath.nPts;
fl.trace("Number of points in the path: " + numOfPoints);
// Displays: Number of points in the path: 0
```



# Hoofdstuk 33: Object presetItem

## Beschikbaarheid

Flash CS4 Professional.

## Beschrijving

Het object `presetItem` vertegenwoordigt een item (voorstelling of map) in het deelvenster Voorinstellingen voor beweging (Venster > Voorinstellingen voor beweging). De array met objecten `presetItem` is een eigenschap van het object `presetPanel` (`presetPanel.items`).

Alle eigenschappen van het object `presetItem` zijn alleen-lezen. Voor taken als verwijderen, naam wijzigen of items verplaatsen gebruikt u de methoden van het [Object presetPanel](#).

## Overzicht van eigenschappen

U kunt de volgende eigenschappen gebruiken met het object `presetItem`:

Eigenschap	Beschrijving
<code>presetItem.isDefault</code>	Hiermee wordt aangegeven of het item samen met Flash is geïnstalleerd of een aangepast item is dat u of iemand anders heeft gemaakt.
<code>presetItem.isFolder</code>	Hiermee wordt aangegeven of het item in het deelvenster Voorinstellingen voor beweging een map of een voorinstelling is.
<code>presetItem.level</code>	Het niveau van het item in de mapstructuur van het deelvenster Voorinstellingen voor beweging.
<code>presetItem.name</code>	De naam van de voorinstelling of map, zonder padinformatie.
<code>presetItem.open</code>	Hiermee wordt aangegeven of een map in het deelvenster Voorinstellingen voor beweging op dat moment is uitgevouwen.
<code>presetItem.path</code>	Het pad naar het item in de mapstructuur van het deelvenster Voorinstellingen voor beweging plus de naam van het item.

## presetItem.isDefault

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetItem.isDefault
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde die opgeeft of het item is geïnstalleerd met Flash (`true`) of dat het een aangepast item is dat u of iemand anders heeft gemaakt (`false`). Als deze waarde `true` is, kunt u het item beschouwen als "alleen-lezen". Het kan niet worden verplaatst of verwijderd en u kunt er geen vergelijkbare bewerkingen op toepassen.

### Voorbeeld

In het volgende voorbeeld wordt de inhoud van het deelvenster Voorinstellingen voor beweging weergegeven en wordt aangegeven of een item met Flash is geïnstalleerd:

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    fl.trace(presetItem.name +", default =" + presetItem.isDefault);
}
```

## presetItem.isFolder

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetItem.isFolder
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde die opgeeft of het item in het deelvenster Voorinstellingen voor beweging een map (`true`) is of een voorinstelling (`false`).

### Voorbeeld

In het volgende voorbeeld wordt weergegeven dat het eerste item in het deelvenster Voorinstellingen voor beweging een map is en het tweede item een voorinstelling:

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].isFolder);
fl.trace(presetItemArray[1].isFolder);
```

## presetItem.level

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetItem.level
```

### Beschrijving

Alleen-lezen eigenschap: een geheel getal dat het niveau van het item in de mapstructuur van het deelvenster Voorinstellingen voor beweging opgeeft. De Standaardmap en de map Eigen voorinstellingen zijn niveau 0.

### Voorbeeld

In het volgende voorbeeld wordt weergegeven dat het eerste item in het deelvenster Voorinstellingen voor beweging niveau 0 is en dat het tweede item niveau 1 is:

```
var presetItemArray=fl.presetPanel.items;  
fl.trace(presetItemArray[0].level);  
fl.trace(presetItemArray[1].level);
```

## presetItem.name

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetItem.name
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die de naam van de voorinstelling of map vertegenwoordigt, zonder padinformatie.

### Voorbeeld

Zie [presetItem.path](#).

## presetItem.open

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetItem.open
```

### Beschrijving

Alleen-lezen eigenschap: hiermee wordt opgegeven of een map in het deelvenster Voorinstellingen voor beweging momenteel is uitgevouwen (`true`) of niet (`false`).

Deze eigenschap is `true` als het item niet een map is. Gebruik [presetItem.isFolder](#) om te bepalen of een item een map of voorinstelling is.

### Voorbeeld

In het volgende voorbeeld wordt weergegeven of mappen in het deelvenster Voorinstellingen voor beweging zijn uitgevouwen of samengevouwen:

```
fl.outputPanel.clear();  
var presetItemArray=fl.presetPanel.items;  
for (i=0;i<presetItemArray.length; i++){  
    var presetItem = presetItemArray[i];  
    if (presetItem.isFolder) {  
        var status = presetItem.open ? "Open" : "Closed"  
        fl.trace(presetItem.level + "-" + presetItem.name + " folder is " + status);  
    }  
}
```

## `presetItem.path`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`presetItem.path`

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks met het pad naar het item in de mapstructuur van het deelvenster Voorinstellingen voor beweging en de naam van het item.

### Voorbeeld

In het volgende voorbeeld wordt het verschil gedemonstreerd tussen de waarden in `presetItem.name` en `presetItem.path`.

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    fl.trace("Name: " + presetItem.name + "\n" + "Path: " + presetItem.path);
    fl.trace("");
}
```

# Hoofdstuk 34: Object presetPanel

## Beschikbaarheid

Flash CS4 Professional.

## Beschrijving

Het object presetPanel vertegenwoordigt het deelvenster Voorinstellingen voor beweging (Venster > Voorinstellingen voor beweging). Het is een eigenschap van het object flash ([fl.presetPanel](#)).

## Overzicht van methoden

U kunt de volgende methoden gebruiken met het object presetPanel:

Methoden	Beschrijving
<code>presetPanel.addNewItem()</code>	Als momenteel één bewegingstween in het werkgebied is geselecteerd, wordt die beweging aan het deelvenster Voorinstellingen voor beweging toegevoegd.
<code>presetPanel.applyPreset()</code>	Hiermee wordt de opgegeven of geselecteerde voorinstelling toegepast op het geselecteerde item in het werkgebied.
<code>presetPanel.deleteFolder()</code>	Hiermee worden de opgegeven map en eventuele submappen verwijderd uit de mapstructuur van het deelvenster Voorinstellingen voor beweging.
<code>presetPanel.deleteItem()</code>	Hiermee wordt de opgegeven voorinstelling verwijderd uit het deelvenster Voorinstellingen voor beweging.
<code>presetPanel.expandFolder()</code>	Hiermee worden een of meer geselecteerde mappen in het deelvenster Voorinstellingen voor beweging uit- of samengevouwen.
<code>presetPanel.exportItem()</code>	Hiermee wordt de momenteel geselecteerde of de opgegeven voorinstelling geëxporteerd naar een XML-bestand.
<code>presetPanel.findItemIndex()</code>	Hiermee wordt een geheel getal geretourneerd dat staat voor de indexlocatie van een item in het deelvenster Voorinstellingen voor beweging.
<code>presetPanel.getSelectedItems()</code>	Hiermee wordt een array met objecten presetItem geretourneerd die overeenstemmen met de momenteel geselecteerde items in het deelvenster Voorinstellingen voor beweging.
<code>presetPanel.importItem()</code>	Hiermee wordt vanuit een opgegeven XML-bestand een voorinstelling toegevoegd aan het deelvenster Voorinstellingen voor beweging.
<code>presetPanel.moveToFolder()</code>	Hiermee wordt het opgegeven item verplaatst naar de opgegeven map.
<code>presetPanel.newFolder()</code>	Hiermee wordt een map gemaakt in de mapstructuur van het deelvenster Voorinstellingen voor beweging.
<code>presetPanel.renameItem()</code>	Hiermee wordt de naam gewijzigd van de geselecteerde voorinstelling of map.
<code>presetPanel.selectItem()</code>	Hiermee wordt een item in het deelvenster Voorinstellingen voor beweging geselecteerd of wordt de selectie opgeheven.

## Overzicht van eigenschappen

U kunt de volgende eigenschap gebruiken met het object presetPanel:

Eigenschap	Beschrijving
<code>presetPanel.items</code>	Een array met objecten <code>presetItem</code> in het deelvenster Voorinstellingen voor beweging.

## `presetPanel.addNewItem()`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
fl.presetPanel.addNewItem( [namePath] );
```

### Parameters

**namePath** Een tekenreeks die het pad en de naam aangeeft van het item dat aan het deelvenster Voorinstellingen voor beweging moet worden toegevoegd. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het item is toegevoegd, anders `false`.

### Beschrijving

Methode; als in het werkgebied één bewegings-tween is geselecteerd, wordt die beweging aan het deelvenster Voorinstellingen voor beweging in de map met de opgegeven naam toegevoegd. Het in *namePath* opgegeven pad moet in het deelvenster staan.

Als een voorinstelling met *namePath* overeenkomt, heeft deze methode geen effect en wordt `false` geretourneerd.

Als u voor *namePath* geen waarde doorgeeft, wordt het item onder de naam “Custom Preset *n*” aan de map Custom Presets toegevoegd. De waarde *n* wordt elke keer dat u een item op deze wijze toevoegt, opgehoogd.

### Voorbeeld

Ervan uitgaande dat in het werkgebied één bewegings-tween is geselecteerd, wordt met de volgende code een voorinstelling Bouncing Ball toegevoegd aan de map Eigen voorinstellingen:

```
fl.presetPanel.addNewItem("Custom Presets/Bouncing Ball");
```

### Zie ook

[presetPanel.newFolder\(\)](#)

## `presetPanel.applyPreset()`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.applyPreset( [presetPath] )
```

**Parameters**

**presetPath** Een tekenreeks die het volledige pad en de naam aangeeft van de voorinstelling die u wilt toepassen, zoals weergegeven in het deelvenster Voorinstellingen voor beweging. Deze parameter is optioneel; als u geen waarde doorgeeft, wordt de momenteel geselecteerde voorinstelling toegepast.

**Geretourneerde waarde**

Een Booleaanse waarde: `true` als de voorinstelling is toegepast, anders `false`.

**Beschrijving**

Methode; hiermee wordt de opgegeven of geselecteerde voorinstelling toegepast op het geselecteerde item in het werkgebied. Het item moet een bewegings-tween zijn, een symbool of een item dat in een symbool kan worden omgezet. Als het item een bewegings-tween is, wordt de huidige beweging vervangen door de geselecteerde voorinstelling. Er verschijnt geen bevestigingsvraag.

Deze methode kan in de volgende omstandigheden niet worden toegepast:

- Het door u als *presetPath* opgegeven pad bestaat niet.
- U geeft geen waarde door voor *presetPath* en er is geen voorinstelling geselecteerd.
- U geeft geen waarde door voor *presetPath* en er zijn meerdere voorinstellingen geselecteerd.
- Het geselecteerde item in het werkgebied is geen symbool en kan worden geconverteerd naar een symbool.

**Voorbeeld**

In het volgende voorbeeld wordt de voorinstelling `aDribble` toegepast op het geselecteerde item in het werkgebied:

```
var result = fl.presetPanel.applyPreset("Custom Presets/Bounces/aDribble");
fl.trace(result);
```

## presetPanel.deleteFolder()

**Beschikbaarheid**

Flash CS4 Professional.

**Gebruik**

```
presetPanel.deleteFolder( [folderPath] )
```

**Parameters**

**folderPath** Een tekenreeks die aangeeft welke map uit het deelvenster Voorinstellingen voor beweging moet worden verwijderd. Deze parameter is optioneel.

**Geretourneerde waarde**

Een Booleaanse waarde: `true` als een of meerdere mappen zijn verwijderd, anders `false`.

**Beschrijving**

Methode; hiermee worden de opgegeven map en eventuele submappen verwijderd uit de mapstructuur van het deelvenster Voorinstellingen voor beweging. Alle voorinstellingen in de mappen worden eveneens verwijderd. U kunt geen mappen verwijderen uit de map Default Presets.

Als u geen waarde doorgeeft voor *folderPath*, worden alle mappen verwijderd die momenteel zijn geselecteerd.

**Opmerking:** De mappen worden zonder controlevraag verwijderd. Deze handeling kan niet ongedaan worden gemaakt.

#### Voorbeeld

Met de onderstaande code wordt een map met de naam `Bouncing` verwijderd uit de map `Eigen` voorinstellingen. Eventuele submappen van `Bouncing` worden ook verwijderd:

```
fl.presetPanel.deleteFolder("Custom Presets/Bouncing");
```

#### Zie ook

[presetPanel.deleteItem\(\)](#)

## presetPanel.deleteItem()

#### Beschikbaarheid

Flash CS4 Professional.

#### Gebruik

```
presetPanel.deleteItem( [namePath] )
```

#### Parameters

**namePath** Een tekenreeks die het pad en de naam aangeeft van het item dat aan het deelvenster Voorinstellingen voor beweging moet worden verwijderd. Deze parameter is optioneel.

#### Geretourneerde waarde

Een Booleaanse waarde: `true` als het item is of de items zijn verwijderd, anders `false`.

#### Beschrijving

Methode; hiermee wordt de opgegeven voorinstelling verwijderd uit het deelvenster Voorinstellingen voor beweging. Als u geen waarde doorgeeft voor *namePath*, worden alle voorinstellingen verwijderd die momenteel zijn geselecteerd. U kunt geen items verwijderen uit de map `Default Presets`.

**Opmerking:** De items worden zonder controlevraag verwijderd. Deze handeling kan niet ongedaan worden gemaakt.

#### Voorbeeld

Met de onderstaande code wordt een voorinstelling met de naam `aDribble` verwijderd uit de map `Eigen` voorinstellingen:

```
fl.presetPanel.deleteItem("Custom Presets/aDribble");
```

#### Zie ook

[presetPanel.deleteFolder\(\)](#)

## presetPanel.expandFolder()

#### Beschikbaarheid

Flash CS4 Professional.



### Gebruik

```
presetPanel.expandFolder( [bExpand [, bRecurse [, folderPath] ] ] )
```

### Parameters

**bExpand** Een Booleaanse waarde die aangeeft of de map moet worden uitgevouwen (`true`) of samengevouwen (`false`). Deze parameter is optioneel; de standaardwaarde is `true`.

**bRecurse** Een Booleaanse waarde die aangeeft of de submappen in de map moeten worden uitgevouwen (`true`) of samengevouwen (`false`). Deze parameter is optioneel; de standaardwaarde is `false`.

**folderPath** Een tekenreeks die het pad aangeeft voor de map die moet worden uit- of samengevouwen. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als een of meer mappen zijn uit- of samengevouwen, anders `false`.

### Beschrijving

Methode; hiermee worden een of meer geselecteerde mappen in het deelvenster Voorinstellingen voor beweging uit- of samengevouwen. Als u andere mappen wilt uit- of samenvouwen dan de mappen die momenteel zijn geselecteerd, geeft u een waarde door voor *folderPath*.

### Voorbeeld

In het volgende voorbeeld wordt de map Custom Presets uitgevouwen, maar de submappen niet:

```
fl.presetPanel.expandFolder(true, false, "Custom Presets");
```

In het volgende voorbeeld worden de map Custom Presets en alle submappen uitgevouwen:

```
fl.presetPanel.expandFolder(true, true, "Custom Presets");
```

## presetPanel.exportItem()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.exportItem(fileURI [, namePath] )
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad en eventueel een bestandsnaam voor het geëxporteerde bestand aangeeft. Zie "Beschrijving" hieronder voor meer informatie.

**namePath** Een tekenreeks die het pad en de naam aangeeft van het item dat in het deelvenster Voorinstellingen voor beweging moet worden geselecteerd. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de voorinstelling is geëxporteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt de momenteel geselecteerde of de opgegeven voorinstelling geëxporteerd naar een XML-bestand. Alleen voorinstellingen kunnen worden geëxporteerd; de methode werkt niet als u een map probeert te exporteren. Deze methode werkt ook niet als u een bestand op schijf probeert te overschrijven.

Als u geen bestandsnaam opgeeft als onderdeel van *fileURI* (het laatste teken van *fileURI* is een slash (/)), wordt het geëxporteerde bestand onder dezelfde naam opgeslagen als de geëxporteerde voorinstelling. Als u geen waarde opgeeft voor *namePath*, wordt de momenteel geselecteerde voorinstelling geëxporteerd. Zie het onderstaande voorbeeld.

### Voorbeeld

In het volgende voorbeeld wordt gedemonstreerd welke bestanden worden gemaakt als verschillende parameters worden doorgegeven met deze methode. U ziet of het opgegeven bestand is gemaakt. Voordat u dit voorbeeld uitvoert, moet u de de voorinstelling `fly-in-left` selecteren in de map `Standaardvoorinstellingen` en de map `My Presets` maken op de schijf.

```
//Exports fly-in-left to C:\My Presets\fly-in-left.xml
fl.presetPanel.exportItem("file:///C:/My Presets/");
//Exports fly-in-left to C:\My Presets\myFavoritePreset.xml
fl.presetPanel.exportItem("file:///C:/My Presets/myFavoritePreset.xml");
// Exports the "pulse" preset to C:\My Presets\pulse.xml
fl.presetPanel.exportItem("file:///C:/My Presets/", "Default Presets/pulse");
// Exports the "pulse" preset to C:\My Presets\thePulsePreset.xml
fl.presetPanel.exportItem("file:///C:/My Presets/thePulsePreset.xml", "Default
Presets/pulse");
```

### Zie ook

[presetPanel.importItem\(\)](#)

## presetPanel.findItemIndex()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.findItemIndex([presetName])
```

### Parameters

**presetName** Een tekenreeks die de naam aangeeft van de voorinstelling waarvoor de indexwaarde wordt geretourneerd. Deze parameter is optioneel.

### Geretourneerde waarde

Een geheel getal dat de index van de opgegeven voorinstelling in de array `presetPanel.items` vertegenwoordigt. Als u geen waarde doorgeeft voor *presetName*, wordt de index van de momenteel geselecteerde voorinstelling geretourneerd. In de volgende situaties retourneert deze methode -1:

- U geeft geen waarde door voor *presetName* en er is geen voorinstelling geselecteerd.
- U geeft geen waarde door voor *presetName* en er zijn meerdere voorinstellingen geselecteerd.
- U geeft een waarde door voor *presetName* die niet overeenstemt met een item in het deelvenster.

### Beschrijving

Methode; hiermee wordt een geheel getal geretourneerd dat staat voor de indexlocatie van een item in het deelvenster Voorinstellingen voor beweging.

### Voorbeeld

Met de volgende code worden de indexwaarde en de volledige padnaam van de momenteel geselecteerde voorinstelling weergegeven:

```
// Select one preset in the Motions Preset panel before running this code
var selectedPreset = fl.presetPanel.findItemIndex();
fl.trace(selectedPreset);
fl.trace(fl.presetPanel.items[selectedPreset].path);
```

## presetPanel.getSelectedItems()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.getSelectedItems()
```

### Parameters

Geen.

### Geretourneerde waarde

Een array van objecten presetItem.

### Beschrijving

Methode; hiermee wordt een array met objecten presetItem geretourneerd die overeenstemmen met de geselecteerde items in het deelvenster Voorinstellingen voor beweging (zie [Object presetItem](#)). Elk item in de array vertegenwoordigt een map of een voorinstelling.

### Voorbeeld

Met de volgende code worden de volledige padnamen van de momenteel geselecteerde items in het deelvenster Voorinstellingen voor beweging weergegeven:

```
var itemArray = fl.presetPanel.getSelectedItems();
var length = itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

### Zie ook

[presetPanel.items](#)

## `presetPanel.importItem()`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.importItem(fileURI [,namePath ])
```

### Parameters

**fileURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het XML-bestand aangeeft dat als voorinstelling in het deelvenster Voorinstellingen voor beweging moet worden geïmporteerd.

**namePath** Een tekenreeks die aangeeft in welke map het geïmporteerde bestand moet worden geplaatst en welke naam het moet krijgen. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het bestand is geïmporteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt vanuit een opgegeven XML-bestand een voorinstelling toegevoegd aan het deelvenster Voorinstellingen voor beweging. Het in *namePath* opgegeven pad moet in het deelvenster staan.

Gebruik `presetPanel.exportItem()` om XML-bestanden te maken die kunnen worden geïmporteerd.

Als u geen waarde doorgeeft voor *namePath*, wordt de geïmporteerde voorinstelling in de map Eigen voorinstellingen geplaatst en krijgt deze dezelfde naam als het geïmporteerde bestand (zonder de XML-extensie).

### Voorbeeld

In het volgende voorbeeld wordt een voorinstelling onder de naam `fastPulse` in de map Eigen voorinstellingen/Pulse geïmporteerd.

```
fl.presetPanel.importItem("file:///C:/My Presets/thePulsePreset.xml", "Custom Presets/Pulse/fastPulse");
```

### Zie ook

[presetPanel.exportItem\(\)](#)

## `presetPanel.items`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.items
```

### Beschrijving

Eigenschap; een array met objecten `presetItem` in het deelvenster Voorinstellingen voor beweging (zie [Object presetItem](#)). Elk item in de array vertegenwoordigt een map of een voorinstelling.

### Voorbeeld

Met de volgende code worden de volledige padnamen van de items in het deelvenster Voorinstellingen voor beweging weergegeven:

```
var itemArray = fl.presetPanel.items;
var length = itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

### Zie ook

[presetPanel.getSelectedItems\(\)](#)

## presetPanel.moveToFolder()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.moveToFolder(folderPath [, namePath] )
```

### Parameters

**folderPath** Een tekenreeks die het pad aangeeft voor de map in het deelvenster Voorinstellingen voor beweging waarnaar een of meerdere items moeten worden verplaatst.

**namePath** Een tekenreeks die het pad en de naam van het te verplaatsen item aangeeft. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de items zijn verplaatst, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven item verplaatst naar de opgegeven map.

Als u een lege tekenreeks ("" ) doorgeeft voor *folderPath*, worden de items verplaatst naar de map Custom Presets. Als u geen waarde doorgeeft voor *namePath*, worden de momenteel geselecteerde items verplaatst.

U kunt geen items uit of naar de map Default Presets verplaatsen.

### Voorbeeld

In het volgende voorbeeld worden de momenteel geselecteerde items verplaatst naar de map Custom Presets/Bouncing. Vervolgens wordt de voorinstelling Fast Bounce naar dezelfde map verplaatst:

```
fl.presetPanel.moveToFolder("Custom Presets/Bouncing");
fl.presetPanel.moveToFolder("Custom Presets/Bouncing" , "Custom Presets/Fast Bounce");
```

## `presetPanel.newFolder()`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.newFolder( [folderPath] )
```

### Parameters

**folderPath** Een tekenreeks die aangeeft waar een nieuwe map in het deelvenster Voorinstellingen voor beweging moet worden toegevoegd, en welke naam de nieuwe map krijgt. Deze parameter is optioneel.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de map is toegevoegd, anders `false`.

### Beschrijving

Methode; hiermee wordt een map gemaakt in de mapstructuur van het deelvenster Voorinstellingen voor beweging. Met deze methode kunt u maar één nieuw mapniveau maken. Dit betekent dat als u "Custom Presets/My First Folder/My Second Folder" doorgeeft voor *folderPath*, "Custom Presets/My First Folder" al moet bestaan in de mapstructuur.

Als u geen waarde doorgeeft voor *folderPath*, wordt op het eerste niveau onder "Custom Presets" een map met de naam "Untitled folder *n*" gemaakt. De waarde *n* wordt elke keer dat u een map op deze wijze toevoegt, opgehoogd.

**Opmerking:** *U kunt geen mappen toevoegen aan de map Default Presets.*

### Voorbeeld

In het volgende voorbeeld wordt een map met de naam `Bouncing` onder de map `Eigen` voorinstellingen toegevoegd:

```
fl.presetPanel.newFolder("Custom Presets/Bouncing");
```

### Zie ook

[presetPanel.addItem\(\)](#)

## `presetPanel.renameItem()`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.renameItem(newName)
```

### Parameters

**newName** Een tekenreeks die de nieuwe naam voor de voorinstelling of map aangeeft.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als de naam van de voorinstelling of de map is gewijzigd, anders `false`.

### Beschrijving

Methode; hiermee wordt de naam gewijzigd van de geselecteerde voorinstelling of map. Deze methode werkt alleen als er in de map Custom Presets maar één voorinstelling of map is geselecteerd. Deze methode kan in de volgende omstandigheden niet worden toegepast:

- Er is geen item geselecteerd.
- Er zijn meerdere items geselecteerd.
- Het geselecteerde item staat in de map Default Presets.
- Op dezelfde locatie als het geselecteerde item staat een item met de naam *newName*.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de geselecteerde voorinstelling in de map Eigen voorinstellingen gewijzigd in `Bounce Faster`.

```
var renamed = fl.presetPanel.renameItem("Bounce Faster");  
fl.trace(renamed);
```

## presetPanel.selectItem()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
presetPanel.selectItem(namePath [, bReplaceCurrentSelection [, bSelect] ])
```

### Parameters

**namePath** Een tekenreeks die het pad en de naam aangeeft van het item dat in het deelvenster Voorinstellingen voor beweging moet worden geselecteerd.

**bReplaceCurrentSelection** Een Booleaanse waarde die aangeeft of het opgegeven item een geselecteerd item vervangt (`true`) of aan het geselecteerde item wordt toegevoegd (`false`). Deze parameter is optioneel; de standaardwaarde is `true`.

**bSelect** Een Booleaanse waarde die aangeeft of het item moet worden geselecteerd (`true`) of dat de selectie moet worden opgeheven (`false`). Deze parameter is optioneel; de standaardwaarde is `true`. Als u `false` doorgeeft voor *bSelect*, wordt de waarde van *bReplaceCurrentSelection* genegeerd.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het item is geselecteerd of de selectie is opgeheven, anders `false`.

### Beschrijving

Methode; hiermee wordt een item geselecteerd in het deelvenster Voorinstellingen voor beweging of wordt de selectie opgeheven. Desgewenst kunnen de momenteel geselecteerde items worden vervangen.

### Voorbeeld

Met de volgende code wordt de voorinstelling `fly-in-blur-right` toegevoegd aan de momenteel geselecteerde voorinstellingen (indien van toepassing) in het deelvenster Voorinstellingen voor beweging:

```
fl.presetPanel.selectItem("Default Presets/fly-in-blur-right", false);
```

# Hoofdstuk 35: Object Rectangle

**Overerving** [Object Element](#) > [Object Shape](#) > Object Rectangle

## Beschikbaarheid

Flash CS3 Professional.

## Beschrijving

Het object Rectangle is een vorm die wordt getekend met het gereedschap Rechthoek primitief. Gebruik [shape.isRectangleObject](#) om te bepalen of een item een object Rectangle is.

## Overzicht van eigenschappen

Naast de eigenschappen van het [Object Shape](#) kunt u de volgende eigenschappen gebruiken voor het object Rectangle. Gebruik [document.setRectangleObjectProperty\(\)](#) om de eigenschappen van een object Rectangle in te stellen.

Eigenschap	Beschrijving
<a href="#">RectangleObject.bottomLeftRadius</a>	Alleen-lezen; een drijvende-kommawaarde die de straal van de linkerbenenhoek van het object Rectangle instelt.
<a href="#">RectangleObject.bottomRightRadius</a>	Alleen-lezen; een drijvende-kommawaarde die de straal van de rechterbenenhoek van het object Rectangle instelt.
<a href="#">RectangleObject.lockFlag</a>	Alleen-lezen; een Booleaanse waarde die bepaalt of verschillende hoeken van de rechthoek verschillende straalwaarden kunnen hebben.
<a href="#">RectangleObject.topLeftRadius</a>	Alleen-lezen; een drijvende-kommawaarde die de straal van alle hoeken van de rechthoek instelt of alleen de straal van de linkerbovenhoek van het object Rectangle.
<a href="#">RectangleObject.topRightRadius</a>	Alleen-lezen; een drijvende-kommawaarde die de straal van de rechterbovenhoek van het object Rectangle instelt.

## RectangleObject.bottomLeftRadius

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`RectangleObject.bottomLeftRadius`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de straal van de linkerbenenhoek van het object Rectangle instelt. Als `RectangleObject.lockFlag` `true` is, heeft het instellen van deze waarde geen effect.

Gebruik [document.setRectangleObjectProperty\(\)](#) om deze waarde in te stellen.



### Zie ook

[document.setRectangleObjectProperty\(\)](#), [RectangleObject.bottomRightRadius](#),  
[RectangleObject.lockFlag](#), [RectangleObject.topLeftRadius](#), [RectangleObject.topRightRadius](#)

## RectangleObject.bottomRightRadius

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`RectangleObject.bottomRightRadius`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de straal van de rechterbenedenhoek van het object `Rectangle` instelt. Als `RectangleObject.lockFlag` `true` is, heeft het instellen van deze waarde geen effect.

Gebruik `document.setRectangleObjectProperty()` om deze waarde in te stellen.

### Zie ook

[document.setRectangleObjectProperty\(\)](#), [RectangleObject.bottomLeftRadius](#),  
[RectangleObject.lockFlag](#), [RectangleObject.topLeftRadius](#), [RectangleObject.topRightRadius](#)

## RectangleObject.lockFlag

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`RectangleObject.lockFlag`

### Beschrijving

Alleen-lezen eigenschap; een Booleaanse waarde die bepaalt of verschillende hoeken van de rechthoek verschillende straalwaarden kunnen hebben. Als deze waarde `true` is, kunnen alle hoeken de waarde hebben die is toegewezen aan `RectangleObject.topLeftRadius`. Als de waarde `false` is, kan de straal van elke hoek onafhankelijk worden ingesteld.

Gebruik `document.setRectangleObjectProperty()` om deze waarde in te stellen.

### Zie ook

[document.setRectangleObjectProperty\(\)](#), [RectangleObject.bottomLeftRadius](#),  
[RectangleObject.bottomRightRadius](#), [RectangleObject.topLeftRadius](#),  
[RectangleObject.topRightRadius](#)

## RectangleObject.topLeftRadius

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`RectangleObject.topLeftRadius`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de straal van alle hoeken van de rechthoek instelt (als `RectangleObject.lockFlag` is `true`) of alleen de straal van de linkerbovenhoek (als `RectangleObject.lockFlag` is `false`).

Gebruik `document.setRectangleObjectProperty()` om deze waarde in te stellen.

### Zie ook

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`,  
`RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topRightRadius`

## RectangleObject.topRightRadius

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`RectangleObject.topRightRadius`

### Beschrijving

Alleen-lezen eigenschap; een drijvende-kommawaarde die de straal van de rechterbovenhoek van het object `Rectangle` instelt. Als `RectangleObject.lockFlag` `true` is, heeft het instellen van deze waarde geen effect.

Gebruik `document.setRectangleObjectProperty()` om deze waarde in te stellen.

### Zie ook

`document.setRectangleObjectProperty()`, `RectangleObject.bottomLeftRadius`,  
`RectangleObject.bottomRightRadius`, `RectangleObject.lockFlag`, `RectangleObject.topLeftRadius`

# Hoofdstuk 36: Object Shape

**Overerving** [Object Element](#) > object Shape

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Shape is een subklasse van het object Element. Het object Shape biedt meer nauwkeurigheid dan de teken-API's bij het bewerken of maken van geometrische vormen in het werkgebied. Dit besturingselement is noodzakelijk, zodat scripts nuttige effecten en andere tekenopdrachten kunnen maken (zie [Object Element](#)).

Alle methoden en eigenschappen van het object Shape die een vorm of onderliggende delen ervan wijzigen moeten tussen aanroepen van `shape.beginEdit()` en `shape.endEdit()` worden geplaatst, anders werken ze niet goed.

## Overzicht van methoden

Naast de methoden van het object Element kunt u de volgende methoden gebruiken voor het object Shape:

Methode	Beschrijving
<code>shape.getCubicSegmentPoints()</code>	Hiermee wordt een array met punten geretourneerd die een kubische curve definiëren.
<code>shape.beginEdit()</code>	Hiermee wordt het begin van een bewerkssessie gedefinieerd.
<code>shape.deleteEdge()</code>	Hiermee wordt de opgegeven rand verwijderd.
<code>shape.endEdit()</code>	Hiermee wordt het einde van een bewerkssessie voor de vorm gedefinieerd.

## Overzicht van eigenschappen

Naast de eigenschappen van het object Element zijn de volgende eigenschappen beschikbaar voor het object Shape:

Eigenschap	Beschrijving
<code>shape.contours</code>	Alleen-lezen; een array met objecten Contour voor de vorm (zie <a href="#">Object Contour</a> ).
<code>shape.edges</code>	Alleen-lezen; een array met objecten Edge (zie <a href="#">Object Edge</a> ).
<code>shape.isDrawingObject</code>	Alleen-lezen; als de waarde true is, is de vorm een tekenobject.
<code>shape.isGroup</code>	Alleen-lezen; als de waarde true is, is de vorm een groep.
<code>shape.isOvalObject</code>	Alleen-lezen; als de waarde true is, is de vorm is een primitief object Oval (gemaakt met het gereedschap Ovaal).
<code>shape.isRectangleObject</code>	Alleen-lezen; als de waarde true is, is de vorm is een primitief object Rectangle (gemaakt met het gereedschap Rechthoek).
<code>shape.members</code>	Een array met objecten in de momenteel geselecteerde groep.
<code>shape.numCubicSegments</code>	Alleen-lezen; het aantal kubische segmenten in de vorm.
<code>shape.vertices</code>	Alleen-lezen; een array met objecten Vertex (zie <a href="#">Object Vertex</a> ).

## shape.beginEdit()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
shape.beginEdit()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het begin van een bewerkssessie gedefinieerd. U moet deze methode gebruiken voordat u een opdracht kiest waarmee het object Shape of een van de onderliggende delen wordt veranderd.

### Voorbeeld

In het volgende voorbeeld wordt de momenteel geselecteerde vorm genomen als invoer en wordt hieruit de eerste rand in de array met randen verwijderd:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

## shape.contours

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
shape.contours
```

### Beschrijving

Alleen-lezen eigenschap; een array van objecten Contour voor de vorm (zie [Object Contour](#)).

### Voorbeeld

In het volgende voorbeeld wordt de eerste contour opgeslagen in de array met contouren in de variabele *c*, vervolgens wordt het [Object HalfEdge](#) van deze contour opgeslagen in de variabele *he*:

```
var c = fl.getDocumentDOM().selection[0].contours[0];  
var he = c.getHalfEdge();
```

## shape.deleteEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
shape.deleteEdge(index)
```

### Parameters

**index** Een op nul gebaseerde index die de rand opgeeft die moet worden verwijderd uit de array [shape.edges](#)shape.edges. Met deze methode wordt de lengte van de array `shape.edges` gewijzigd.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de opgegeven rand verwijderd. U moet [shape.beginEdit\(\)](#) aanroepen voordat u deze methode gebruikt.

### Voorbeeld

In het volgende voorbeeld wordt de momenteel geselecteerde vorm genomen als invoer en wordt eerste rand in de array met randen verwijderd:

```
var shape = fl.getDocumentDOM().selection[0];  
shape.beginEdit();  
shape.deleteEdge(0);  
shape.endEdit();
```

## shape.edges

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
shape.edges
```

### Beschrijving

Alleen-lezen eigenschap; een array van objecten [Edge](#) (zie [Object Edge](#)).

## shape.endEdit()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
shape.endEdit()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het einde van een bewerkssessie voor de vorm gedefinieerd. Alle wijzigingen die zijn aangebracht in het object Shape of een van de onderliggende delen, worden toegepast op de vorm. U moet deze methode gebruiken nadat u een opdracht kiest waarmee het object Shape of een van de onderliggende delen wordt veranderd.

### Voorbeeld

In het volgende voorbeeld wordt de momenteel geselecteerde vorm genomen als invoer en wordt hieruit de eerste rand in de array met randen verwijderd:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

## shape.getCubicSegmentPoints()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
shape.getCubicSegmentPoints(cubicSegmentIndex)
```

### Parameters

**cubicSegmentIndex** Een geheel getal dat het cubische segment opgeeft waarvoor punten worden geretourneerd.

### Geretourneerde waarde

Een array met punten die een cubische curve definieert voor het object Edge die overeenkomt met de opgegeven *cubicSegmentIndex* (zie [edge.cubicSegmentIndex](#)).

### Beschrijving

Methode; hiermee wordt een array met punten geretourneerd die een kubische curve definiëren.

### Voorbeeld

In het volgende voorbeeld worden de *x*- en *y*-waarde weergegeven voor elk punt op de cubische curve van de eerste rand van de selectie:

```
var elem = fl.getDocumentDOM().selection[0];
var index = elem.edges[0].cubicSegmentIndex;
var cubicPoints = elem.getCubicSegmentPoints(index);
for (i=0; i<cubicPoints.length; i++) {
    fl.trace("index " + i + " x: " + cubicPoints[i].x + " y: " + cubicPoints[i].y);
}
```

## shape.isDrawingObject

### Beschikbaarheid

Flash 8.

### Gebruik

```
shape.isDrawingObject
```

### Beschrijving

Alleen-lezen eigenschap; als de waarde `true` is, is de vorm een tekenobject.

### Voorbeeld

In het volgende voorbeeld wordt het eerste geselecteerde object opgeslagen in de variabele `sel`, vervolgens wordt aan de hand van de eigenschappen `element.elementType` en `shape.isDrawingObject` bepaald of het geselecteerde object een tekenobject is:

```
var sel = fl.getDocumentDOM().selection[0];
var shapeDrawingObject = (sel.elementType == "shape") && sel.isDrawingObject;
fl.trace(shapeDrawingObject);
```

### Zie ook

[document.crop\(\)](#), [document.deleteEnvelope\(\)](#), [document.intersect\(\)](#), [document.punch\(\)](#),  
[document.union\(\)](#), [shape.isGroup](#)

## shape.isGroup

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
shape.isGroup
```

### Beschrijving

Alleen-lezen eigenschap; als de waarde `true` is, is de vorm een groep. Een groep kan elementen van verschillende typen bevatten, zoals tekstelementen en symbolen. De groep zelf wordt echter als een vorm beschouwd en u kunt de eigenschap `shape.isGroup` gebruiken, ongeacht het type elementen dat de groep bevat.

### Voorbeeld

In het volgende voorbeeld wordt het eerste geselecteerde object opgeslagen in de variabele `sel`, vervolgens wordt aan de hand van de eigenschappen `element.elementType` en `shape.isGroup` bepaald of het geselecteerde object een groep is:

```
var sel = fl.getDocumentDOM().selection[0];  
var shapeGroup = (sel.elementType == "shape") && sel.isGroup;  
fl.trace(shapeGroup);
```

### Zie ook

[shape.isDrawingObject](#)

## shape.isOvalObject

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`shape.isOvalObject`

### Beschrijving

Alleen-lezen eigenschap; als de waarde `true` is, is de vorm is een primitief object Oval (gemaakt met het gereedschap Ovaal primitief).

### Voorbeeld

In het volgende voorbeeld wordt "true" weergegeven als het eerste geselecteerde item een primitief object Oval is en "false" als dit niet zo is:

```
var sel = fl.getDocumentDOM().selection[0];  
fl.trace(sel.isOvalObject);
```

### Zie ook

[shape.isRectangleObject](#)

## shape.isRectangleObject

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

`shape.isRectangleObject`

### Beschrijving

Alleen-lezen eigenschap; als de waarde `true` is, is de vorm is een primitief object Rectangle (gemaakt met het gereedschap Rechthoek primitief).



### Voorbeeld

In het volgende voorbeeld wordt "true" weergegeven als het eerste geselecteerde item een primitief object Rectangle is en "false" als dit niet zo is:

```
var sel = fl.getDocumentDOM().selection[0];  
fl.trace(sel.isRectangleObject);
```

### Zie ook

[shape.isOvalObject](#)

## shape.members

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`shape.members`

### Beschrijving

Alleen-lezen eigenschap; een array met objecten in de momenteel geselecteerde groep. Deze eigenschap is alleen beschikbaar als de waarde van `shape.isGrouptrue` is. Onbewerkte vormen in de groep worden niet opgenomen in de array `shape.members`.

Als de groep bijvoorbeeld drie tekenobjecten bevat en drie onbewerkte vormen, bevat de array `shape.members` drie elementen, één voor elk tekenobject. Als de groep alleen onbewerkte vormen bevat, is de array leeg.

### Voorbeeld

Met de volgende code wordt het aantal cubische segmenten van elk tekenobject in de momenteel geselecteerde groep weergegeven:

```
var shapesArray = fl.getDocumentDOM().selection[0].members;  
for (i=0; i<shapesArray.length; i++) {  
    fl.trace(shapesArray[i].numCubicSegments);  
}
```

### Zie ook

[shape.isGroup](#)

## shape.numCubicSegments

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

`shape.numCubicSegments`

### Beschrijving

Alleen-lezen eigenschap; het aantal cubische segmenten in de vorm.

### Voorbeeld

Ervan uitgaande dat er een vierkant of rechthoek is geselecteerd, wordt met de volgende code "4" weergegeven in het deelvenster Uitvoer:

```
var theShape = fl.getDocumentDOM().selection[0];  
fl.trace(theShape.numCubicSegments);
```

## shape.vertices

### Beschikbaarheid

Flash MX 2004.

### Gebruik

shape.vertices

### Beschrijving

Alleen-lezen eigenschap; een array van objecten [Vertex](#) (zie [Object Vertex](#)).

### Voorbeeld

In het volgende voorbeeld wordt het eerste geselecteerde object opgeslagen in de variabele `someShape`, vervolgens wordt het aantal hoekpunten voor dat object getoond in het deelvenster Uitvoer:

```
var someShape = fl.getDocumentDOM().selection[0];  
fl.trace("The shape has " + someShape.vertices.length + " vertices.");
```

# Hoofdstuk 37: Object SoundItem

**Overerving** [Object Item](#) > object SoundItem

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object SoundItem is een subklasse van het object Item. Het staat voor een bibliotheekitem dat wordt gebruikt om een geluid te maken. Zie ook [frame.soundLibraryItem](#) en [Object Item](#).

## Overzicht van methoden

Het object SoundItem heeft naast de methoden voor het object Item ook de volgende methode:

Eigenschap	Beschrijving
<code>soundItem.exportToFile()</code>	Hiermee wordt het opgegeven bestand geëxporteerd naar een QuickTime-bestand op de Macintosh, of naar een WAV- of QT-bestand onder Windows.

## Overzicht van eigenschappen

Naast de eigenschappen van het object Item zijn de volgende eigenschappen beschikbaar voor het object SoundItem:

Eigenschap	Beschrijving
<code>soundItem.bitRate</code>	Een tekenreeks die de bitsnelheid van een geluid in de bibliotheek aangeeft. Alleen beschikbaar voor het compressietype MP3.
<code>soundItem.bits</code>	Een tekenreeks die de bitwaarde voor een geluid in de bibliotheek met ADPCM-compressie aangeeft.
<code>soundItem.compressionType</code>	Een tekenreeks die het compressietype voor een geluid in de bibliotheek aangeeft.
<code>soundItem.convertStereoToMono</code>	Een Booleaanse waarde die alleen beschikbaar is voor de compressietypen MP3 en Raw.
<code>soundItem.fileLastModifiedDate</code>	Alleen-lezen; een tekenreeks die een hexadecimaal getal bevat dat het aantal seconden vertegenwoordigt dat is verstreken tussen 1 januari 1970 en de wijzigingsdatum van het oorspronkelijke bestand (op schijf) op het moment dat het bestand is geïmporteerd naar de bibliotheek.
<code>soundItem.originalCompressionType</code>	Alleen-lezen; een tekenreeks die opgeeft of het opgegeven item is geïmporteerd als MP3-bestand.
<code>soundItem.quality</code>	Een tekenreeks die de weergavekwaliteit van een geluid in de bibliotheek aangeeft. Alleen beschikbaar voor het compressietype MP3.
<code>soundItem.sampleRate</code>	Een tekenreeks die de samplingfrequentie voor de audioclip aangeeft.
<code>soundItem.sourceFileExists</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of het bestand dat naar de bibliotheek is geïmporteerd, nog bestaat op de locatie vanwaar het werd geïmporteerd.

Eigenschap	Beschrijving
<code>soundItem.sourceFileIsCurrent</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of de wijzigingsdatum van het bibliotheekitem gelijk is aan de wijzigingsdatum (op schijf) van het bestand dat is geïmporteerd.
<code>soundItem.sourceFilePath</code>	Alleen-lezen; een tekenreeks, uitgedrukt als bestand:/// URI, die staat voor het pad en de naam van het bestand dat naar de bibliotheek is geïmporteerd.
<code>soundItem.useImportedMP3Quality</code>	Een Booleaanse waarde; bij <code>true</code> worden alle andere eigenschappen genegeerd en wordt de geïmporteerde MP3-kwaliteit gebruikt.

## soundItem.bitRate

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`soundItem.bitRate`

### Beschrijving

Eigenschap; een tekenreeks die de bitsnelheid van een geluid in de bibliotheek opgeeft. Deze eigenschap is alleen beschikbaar voor het compressietype MP3. Acceptabele waarden zijn "8 kbps", "16 kbps", "20 kbps", "24 kbps", "32 kbps", "48 kbps", "56 kbps", "64 kbps", "80 kbps", "112 kbps", "128 kbps" en "160 kbps". Stereogeluiden die met 8 Kbps of 16 Kbps worden geëxporteerd, worden tot geconverteerd naar mono. De eigenschap is `undefined` voor andere compressietypen.

Stel `soundItem.useImportedMP3Quality` in op `false` als u een waarde voor deze eigenschap wilt opgeven.

### Voorbeeld

In het volgende voorbeeld wordt de waarde `bitRate` in het deelvenster Uitvoer weergegeven als het opgegeven item in de bibliotheek het compressietype MP3 heeft:

```
alert(fl.getDocumentDOM().library.items[0].bitRate);
```

### Zie ook

[soundItem.compressionType](#), [soundItem.convertStereoToMono](#)

## soundItem.bits

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`soundItem.bits`

### Beschrijving

Eigenschap; een tekenreeks die de bitwaarde voor een geluid in de bibliotheek met ADPCM-compressie aangeeft. Acceptabele waarden zijn "2 bit", "3 bit", "4 bit" en "5 bit".

Stel `soundItem.useImportedMP3Quality` in op `false` als u een waarde voor deze eigenschap wilt opgeven.

### Voorbeeld

In het volgende voorbeeld wordt de bitwaarde in het deelvenster Uitvoer weergegeven als het momenteel geselecteerde item in de bibliotheek het compressietype ADPCM heeft:

```
alert(fl.getDocumentDOM().library.items[0].bits);
```

### Zie ook

[soundItem.compressionType](#)

## soundItem.compressionType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
soundItem.compressionType
```

### Beschrijving

Eigenschap; een tekenreeks die het compressietype voor een geluid in de bibliotheek aangeeft. Acceptabele waarden zijn "Default", "ADPCM", "MP3", "Raw" en "Speech".

Stel `soundItem.useImportedMP3Quality` in op `false` als u een waarde voor deze eigenschap wilt opgeven.

### Voorbeeld

In het volgende voorbeeld wordt een item in de bibliotheek gewijzigd in het compressietype Raw:

```
fl.getDocumentDOM().library.items[0].compressionType = "Raw";
```

In het volgende voorbeeld wordt het compressietype van de geselecteerde bibliotheekitems gewijzigd in Speech:

```
fl.getDocumentDOM().library.getSelectedItems().compressionType = "Speech";
```

### Zie ook

[soundItem.originalCompressionType](#)

## soundItem.convertStereoToMono

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
soundItem.convertStereoToMono
```

### Beschrijving

Eigenschap; een Booleaanse waarde die alleen beschikbaar is voor de compressietypen MP3 en Raw. Door deze waarde op `true` in te stellen maakt u een stereogeluid mono; bij `false` blijft het stereogeluid behouden. Als bij het compressietype MP3 `soundItem.bitRate` minder is dan 20 Kbps, wordt deze eigenschap genegeerd en wordt automatisch `true` ingesteld (zie `soundItem.bitRate`).

Stel `soundItem.useImportedMP3Quality` in op `false` als u een waarde voor deze eigenschap wilt opgeven.

### Voorbeeld

In het volgende voorbeeld wordt een item in de bibliotheek geconverteerd naar mono als het compressietype MP3 of Raw is:

```
fl.getDocumentDOM().library.items[0].convertStereoToMono = true;
```

### Zie ook

`soundItem.compressionType`

## `soundItem.exportToFile()`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
soundItem.exportToFile(fileURI)
```

### Parameters

`fileURI` Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad en de naam van het geëxporteerde bestand aangeeft.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het bestand is geëxporteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven bestand geëxporteerd naar een QuickTime-bestand op de Macintosh, of naar een WAV- of QT-bestand onder Windows. Het geëxporteerde QuickTime- of QT-bestand bevat alleen audio; video wordt niet geëxporteerd. De exportinstellingen zijn afhankelijk van het item dat wordt geëxporteerd.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een geluidsitem is, wordt dit item met de volgende code geëxporteerd als WAV-bestand:

```
var soundFileURL = "file:///C:/out.wav";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFile(soundFileURL);
```

## soundItem.fileLastModifiedDate

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
soundItem.fileLastModifiedDate
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die een hexadecimaal getal bevat dat het aantal seconden vertegenwoordigt dat is verstreken tussen 1 januari 1970 en de wijzigingsdatum van het oorspronkelijke bestand (op schijf) op het moment dat het bestand werd geïmporteerd naar de bibliotheek. Als het bestand niet meer bestaat, is deze waarde "00000000".

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een geluidsitem is, wordt met de volgende code een hexadecimaal getal weergegeven zoals hierboven beschreven.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

### Zie ook

[soundItem.sourceFileExists](#), [soundItem.sourceFileIsCurrent](#), [soundItem.sourceFilePath](#),  
[FLfile.getModificationDate\(\)](#)

## soundItem.originalCompressionType

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
soundItem.originalCompressionType
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die opgeeft of het opgegeven item is geïmporteerd als MP3-bestand. Mogelijke waarden voor deze eigenschap zijn "RAW" en "MP3".

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een geluidsitem is, wordt met de volgende code "MP3" weergegeven als het bestand als MP3-bestand naar de bibliotheek is geïmporteerd, en "RAW" als dit niet het geval is:

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Imported compression type = " + libItem.originalCompressionType);
```

### Zie ook

[soundItem.compressionType](#)

## soundItem.quality

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`soundItem.quality`

### Beschrijving

Eigenschap; een tekenreeks die de weergavekwaliteit van een geluid in de bibliotheek aangeeft. Deze eigenschap is alleen beschikbaar voor het compressietype MP3. Acceptabele waarden zijn "Fast", "Medium" en "Best".

Stel `soundItem.useImportedMP3Quality` in op `false` als u een waarde voor deze eigenschap wilt opgeven.

### Voorbeeld

In het volgende voorbeeld wordt de weergavekwaliteit van een item in de bibliotheek ingesteld op `Best` als het compressietype MP3 is:

```
fl.getDocumentDOM().library.items[0].quality = "Best";
```

### Zie ook

[soundItem.compressionType](#)

## soundItem.sampleRate

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`soundItem.sampleRate`

### Beschrijving

Eigenschap; een tekenreeks die de bemonsteringsfrequentie voor de audioclip aangeeft. Deze eigenschap is alleen beschikbaar voor de compressietypen ADPCM, Raw en Speech. Acceptabele waarden zijn "5 kHz", "11 kHz", "22 kHz" en "44 kHz".

Stel `soundItem.useImportedMP3Quality` in op `false` als u een waarde voor deze eigenschap wilt opgeven.

### Voorbeeld

In het volgende voorbeeld wordt de samplingfrequentie van een item in de bibliotheek ingesteld op 5 kHz als het compressietype ADPCM, Raw of Speech is:

```
fl.getDocumentDOM().library.items[0].sampleRate = "5 kHz";
```

### Zie ook

[soundItem.compressionType](#)



## `soundItem.sourceFileExists`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
soundItem.sourceFileExists
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde: `true` als het bestand dat werd geïmporteerd in de bibliotheek, nog bestaat op de locatie vanwaar het werd geïmporteerd, anders `false`.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een geluidsitem is, wordt met de volgende code "true" weergegeven als het bestand dat naar de bibliotheek werd geïmporteerd, nog bestaat.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

### Zie ook

[soundItem.sourceFileIsCurrent](#), [soundItem.sourceFilePath](#)

## `soundItem.sourceFileIsCurrent`

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
soundItem.sourceFileIsCurrent
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde: `true` als de wijzigingsdatum van het bibliotheekitem gelijk is aan de wijzigingsdatum op schijf van het bestand dat werd geïmporteerd, anders `false`.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een geluidsitem is, wordt met de volgende code "true" weergegeven als het bestand dat werd geïmporteerd, niet op schijf is gewijzigd sinds het werd geïmporteerd.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

### Zie ook

[soundItem.fileLastModifiedDate](#), [soundItem.sourceFilePath](#)

## soundItem.sourceFilePath

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
soundItem.sourceFilePath
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks, uitgedrukt als bestand:/// URI, die het pad en de naam vertegenwoordigt van het bestand dat naar de bibliotheek werd geïmporteerd.

### Voorbeeld

In het onderstaande voorbeeld ziet u de naam en het pad naar het bronbestand van items in de bibliotheek van het type "video":

```
for (idx in fl.getDocumentDOM().library.items) {  
  if (fl.getDocumentDOM().library.items[idx].itemType == "sound") {  
    var myItem = fl.getDocumentDOM().library.items[idx];  
    fl.trace(myItem.name + " source is " + myItem.sourceFilePath);  
  }  
}
```

### Zie ook

[soundItem.sourceFileExists](#)

## soundItem.useImportedMP3Quality

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
soundItem.useImportedMP3Quality
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Bij `true` worden alle andere eigenschappen genegeerd, en wordt de geïmporteerde MP3-kwaliteit gebruikt.

### Voorbeeld

In het volgende voorbeeld wordt een item in de bibliotheek geplaatst, zodat de geïmporteerde MP3-kwaliteit kan worden gebruikt:

```
fl.getDocumentDOM().library.items[0].useImportedMP3Quality = true;
```

### Zie ook

[soundItem.compressionType](#)

# Hoofdstuk 38: Object Stroke

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Stroke bevat alle instellingen voor een streek, waaronder de aangepaste instellingen. Dit object vertegenwoordigt de informatie in de Eigenschapcontrole. Als u het object Stroke samen met de methode `document.setCustomStroke()` gebruikt, kunt u de streekinstellingen voor het deelvenster Gereedschappen, de Eigenschapcontrole en de huidige selectie wijzigen. Verder kunt u de streekinstellingen van het deelvenster Gereedschappen en de Eigenschapcontrole of van de huidige selectie ook ophalen met de methode `document.getCustomStroke()`.

Dit object heeft altijd de volgende vier eigenschappen: `style`, `thickness`, `color` en `breakAtCorners`. (In Flash CS3 is de eigenschap `breakAtCorners` vervangen door `stroke.joinType`.) Afhankelijk van de waarde van de eigenschap `stroke.style` kunnen ook andere eigenschappen worden ingesteld.

## Overzicht van eigenschappen

Voor het object Stroke zijn de volgende eigenschappen beschikbaar:

Eigenschap	Beschrijving
<code>stroke.breakAtCorners</code>	Een Booleaanse waarde, is gelijk aan de instelling Scherpe hoeken in het aangepaste dialoogvenster Streekstijl.
<code>stroke.capType</code>	Een tekenreeks die het type uiteinde voor de streek aangeeft.
<code>stroke.color</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat de streekkleur vertegenwoordigt.
<code>stroke.curve</code>	Een tekenreeks die het type arcering voor de streek aangeeft.
<code>stroke.dash1</code>	Een geheel getal dat de lengte van het effen gedeelte in een onderbroken lijn aangeeft.
<code>stroke.dash2</code>	Een geheel getal dat de lengte van het lege gedeelte in een onderbroken lijn aangeeft.
<code>stroke.density</code>	Een tekenreeks die de dichtheid van een gespikkelde lijn aangeeft.
<code>stroke.dotSize</code>	Een tekenreeks die de stipgrootte van een gespikkelde lijn aangeeft.
<code>stroke.dotSpace</code>	Een geheel getal dat de afstand tussen stippen in een gestippelde lijn aangeeft.
<code>stroke.hatchThickness</code>	Een tekenreeks die de dikte van een gearceerde lijn aangeeft.
<code>stroke.jiggle</code>	Een tekenreeks die de schommeling van een gearceerde lijn aangeeft.
<code>stroke.joinType</code>	Een tekenreeks die het type samenvoeging voor de streek aangeeft.
<code>stroke.length</code>	Een tekenreeks die de lengte van een gearceerde lijn aangeeft.
<code>stroke.miterLimit</code>	Een drijvende-kommawaarde die de hoek aangeeft vanaf wanneer de top van het verstek door een segment wordt afgevlakt.
<code>stroke.pattern</code>	Een tekenreeks die het patroon van een ongelijke lijn aangeeft.
<code>stroke.rotate</code>	Een tekenreeks die de rotatie van een gearceerde lijn aangeeft.
<code>stroke.scaleType</code>	Een tekenreeks die de op de streek toe te passen schaal aangeeft.

Eigenschap	Beschrijving
<code>stroke.shapeFill</code>	Een <b>Object Fill</b> dat staat voor de vulinstellingen van de streek.
<code>stroke.space</code>	Een tekenreeks die de afstand van een gearceerde lijn aangeeft.
<code>stroke.strokeHinting</code>	Een Booleaanse waarde die aangeeft of er op de streek streekhints staan ingesteld.
<code>stroke.style</code>	Een tekenreeks die de streekstijl beschrijft.
<code>stroke.thickness</code>	Een geheel getal dat de streekgrootte aangeeft.
<code>stroke.variation</code>	Een tekenreeks die de variatie van een gespikkelde lijn aangeeft.
<code>stroke.waveHeight</code>	Een tekenreeks die de golfhoogte van een ongelijke lijn aangeeft.
<code>stroke.waveLength</code>	Een tekenreeks die de golflengte van een ongelijke lijn aangeeft.

## stroke.breakAtCorners

### Beschikbaarheid

Flash MX 2004. In Flash CS3 afgekeurd en vervangen door `stroke.joinType`.

### Gebruik

```
stroke.breakAtCorners
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Deze eigenschap is gelijk aan de instelling Scherpe hoeken in het aangepaste dialoogvenster Streekstijl.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `breakAtCorners` ingesteld op `true`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.breakAtCorners = true;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.capType

### Beschikbaarheid

Flash 8.

### Gebruik

```
stroke.capType
```

### Beschrijving

Eigenschap; een tekenreeks die het type uiteinde voor de streek aangeeft. Acceptabele waarden zijn "none", "round" en "square".

### Voorbeeld

In het volgende voorbeeld wordt het type streekuiteinde ingesteld op `round`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.capType = "round";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.color

### Beschikbaarheid

Flash MX 2004. (In Flash 8 en hoger is deze eigenschap afgekeurd en vervangen door `stroke.shapeFill.color`).

### Gebruik

`stroke.color`

### Beschrijving

Eigenschap; de kleur van de streek, in één van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

### Voorbeeld

In het volgende voorbeeld wordt de streekkleur ingesteld:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.color = "#000000";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

### Zie ook

[stroke.shapeFill](#)

## stroke.curve

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.curve`

### Beschrijving

Eigenschap; een tekenreeks die het type arcering voor de streek aangeeft. Deze eigenschap kan alleen worden ingesteld wanneer de eigenschap `stroke.style` is ingesteld op "hatched" (zie [stroke.style](#)). Acceptabele waarden zijn "straight", "slight curve", "medium curve" en "very curved".

### Voorbeeld

In het volgende voorbeeld worden de `curve`-eigenschap en andere eigenschappen ingesteld voor een streek met de stijl `hatched`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.dash1

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
stroke.dash1
```

### Beschrijving

Eigenschap; een geheel getal dat de lengte van de effen gedeelten in een onderbroken lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `dashed` (zie [stroke.style](#)).

### Voorbeeld

In het volgende voorbeeld worden de eigenschappen `dash1` en `dash2` ingesteld voor de streekstijl `dashed`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "dashed";
myStroke.dash1 = 1;
myStroke.dash2 = 2;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.dash2

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
stroke.dash2
```

### Beschrijving

Eigenschap; een geheel getal dat de lengte van de lege gedeelten in een onderbroken lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `dashed` (zie [stroke.style](#)).

### Voorbeeld

Zie [stroke.dash1](#).

## stroke.density

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.density`

### Beschrijving

Eigenschap; een tekenreeks die de dichtheid van een gespikkelde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `stipple` (zie [stroke.style](#)). Acceptabele waarden zijn "very dense", "dense", "sparse" en "very sparse".

### Voorbeeld

In het volgende voorbeeld wordt de dichtheid voor de streekstijl `stipple` ingesteld op `sparse`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.dotSize

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.dotSize`

### Beschrijving

Eigenschap; een tekenreeks die de stipgrootte van een gespikkelde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `stipple` (zie [stroke.style](#)). Acceptabele waarden zijn "tiny", "small", "medium" en "large".

In het volgende voorbeeld wordt de eigenschap `dotSize` voor de streekstijl `stipple` ingesteld op `tiny`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.dotsize = "tiny";
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.dotSpace

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.dotSpace`

### Beschrijving

Eigenschap; een geheel getal dat de afstand tussen stippen in een gestippelde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `dotted`. Zie [stroke.style](#).

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `dotSpace` voor een streekstijl `dotted` ingesteld op 3:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "dotted";
myStroke.dotSpace = 3;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.hatchThickness

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.hatchThickness`

### Beschrijving

Eigenschap; een tekenreeks die de dikte van een gearceerde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `hatched` (zie [stroke.style](#)). Acceptabele waarden zijn "hairline", "thin", "medium" en "thick".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `hatchThickness` voor de streekstijl `hatched` ingesteld op `thin`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```



## stroke.jiggle

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.jiggle`

### Beschrijving

Eigenschap; een tekenreeks die de schommeling van een gearceerde lijn opgeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `hatched` (zie [stroke.style](#)). Acceptabele waarden zijn "none", "bounce", "loose" en "wild".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `jiggle` voor de streekstijl `hatched` ingesteld op `wild`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.joinType

### Beschikbaarheid

Flash 8.

### Gebruik

`stroke.joinType`

### Beschrijving

Eigenschap; een tekenreeks die het type samenvoeging voor de streek aangeeft. Acceptabele waarden zijn "miter", "round" en "bevel".

### Zie ook

[stroke.capType](#)

## stroke.length

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.length`

### Beschrijving

Eigenschap; een tekenreeks die de lengte van een gearceerde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `hatched` (zie [stroke.style](#)). Acceptabele waarden zijn "equal", "slight variation", "medium variation" en "random" (de waarde "random" is toegewezen aan "medium variation".)

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `length` voor de streekstijl `hatched` ingesteld op `slight`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight variation";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.miterLimit

### Beschikbaarheid

Flash 8.

### Gebruik

`stroke.miterLimit`

### Beschrijving

Eigenschap; een drijvende-kommawaarde die de hoek aangeeft vanaf wanneer de top van het verstek door een segment wordt afgevlakt. Dit betekent dat het verstek enkel wordt afgevlakt als de hoek ervan groter is dan de waarde voor `miterLimit`.

### Voorbeeld

In het volgende voorbeeld wordt de verstekgrens van de streek gewijzigd in 3. Wanneer de hoek van het verstek groter is dan 3, wordt dit afgevlakt.

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.miterLimit = 3;
var myStroke = fl.getDocumentDOM().setCustomStroke();
```

## stroke.pattern

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.pattern`

### Beschrijving

Eigenschap; een tekenreeks die het patroon van een ongelijke lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `ragged` (zie [stroke.style](#)). Acceptabele waarden zijn "solid", "simple", "random", "dotted", "random dotted", "triple dotted" en "random triple dotted".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `pattern` voor de streekstijl `ragged` ingesteld op `random`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.rotate

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.rotate`

### Beschrijving

Eigenschap; een tekenreeks die de rotatie van een gearceerde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `hatched` (zie [stroke.style](#)). Acceptabele waarden zijn "none", "slight", "medium" en "free".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `rotate` voor de streekstijl `hatched` ingesteld op `free`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "hatched";  
myStroke.curve = "straight";  
myStroke.space = "close";  
myStroke.jiggle = "wild";  
myStroke.rotate = "free";  
myStroke.length = "slight";  
myStroke.hatchThickness = "thin";
```

## stroke.scaleType

### Beschikbaarheid

Flash 8.

### Gebruik

`stroke.scaleType`

### Beschrijving

Eigenschap; een tekenreeks die de op de streek toe te passen schaal aangeeft. Acceptabele waarden zijn "normal", "horizontal", "vertical" en "none".

### Voorbeeld

In het volgende voorbeeld wordt het schaaltype van de streek ingesteld op `horizontal`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.scaleType = "horizontal";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.shapeFill

### Beschikbaarheid

Flash 8.

### Gebruik

`stroke.shapeFill`

### Beschrijving

Eigenschap; een [Object Fill](#) dat de vulinstellingen van de streek vertegenwoordigt.

### Voorbeeld

In het volgende voorbeeld worden vulinstellingen aangegeven en op de streek toegepast:

```
var fill = fl.getDocumentDOM().getCustomFill();  
fill.linearGradient = true;  
fill.colorArray = [ 00ff00, ff0000, fffff ];  
var stroke = fl.getDocumentDOM().getCustomStroke();  
stroke.shapeFill = fill;  
fl.getDocumentDOM().setCustomStroke(stroke);
```

## stroke.space

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`stroke.space`

### Beschrijving

Eigenschap; een tekenreeks die de afstand van een gearceerde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `hatched` (zie [stroke.style](#)). Acceptabele waarden zijn "very close", "close", "distant" en "very distant".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `space` voor de streekstijl `hatched` ingesteld op `close`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.strokeHinting

### Beschikbaarheid

Flash 8.

### Gebruik

```
stroke.strokeHinting
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of er op de streek streekhints staan ingesteld.

### Voorbeeld

In het volgende voorbeeld worden de streekhints voor de streek geactiveerd:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.strokeHinting = true;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.style

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
stroke.style
```

### Beschrijving

Eigenschap; een tekenreeks die de streekstijl beschrijft. Acceptabele waarden zijn `"noStroke"`, `"solid"`, `"dashed"`, `"dotted"`, `"ragged"`, `"stipple"` en `"hatched"`. Voor een aantal van deze waarden moeten in het object `Stroke` aanvullende eigenschappen worden ingesteld, zoals beschreven in onderstaande lijst:

- Voor de waarde `"solid"` of `"noStroke"` zijn er geen andere eigenschappen.
- Voor de waarde `dashed` zijn er twee extra eigenschappen: `dash1` en `dash2`.
- Voor de waarde `dotted` is er één extra eigenschap: `dotSpace`.

- Voor de waarde `ragged` zijn er drie extra eigenschappen: `pattern`, `waveHeight` en `waveLength`.
- Voor de waarde `stipple` zijn er drie extra eigenschappen: `dotSize`, `variation` en `density`.
- Voor de waarde `hatched` zijn er zes extra eigenschappen: `hatchThickness`, `space`, `jiggle`, `rotate`, `curve` en `length`.

### Voorbeeld

In het volgende voorbeeld wordt de streekstijl ingesteld op `ragged`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.thickness

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
stroke.thickness
```

### Beschrijving

Eigenschap; een geheel getal dat de streekgrootte opgeeft.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `thickness` van de streek ingesteld op de waarde 2:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.thickness = 2;  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.variation

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
stroke.variation
```

### Beschrijving

Eigenschap; een tekenreeks die de variatie van een gespikkelde lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `stipple` (zie [stroke.style](#)). Acceptabele waarden zijn "one size", "small variation", "varied sizes" en "random sizes".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `variation` voor een streekstijl `stipple` ingesteld op `random sizes`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "stipple";  
myStroke.dotSpace= 3;  
myStroke.variation = "random sizes";  
myStroke.density = "sparse";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.waveHeight

### Beschikbaarheid

Flash MX 2004.

### Gebruik

stroke.waveHeight

### Beschrijving

Eigenschap; een tekenreeks die de golfhoogte van een ongelijke lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `ragged` (zie [stroke.style](#)). Acceptabele waarden zijn "flat", "wavy", "very wavy" en "wild".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `waveHeight` voor de streekstijl `ragged` ingesteld op `flat`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
myStroke.waveHeight = "flat";  
myStroke.waveLength = "short";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```

## stroke.waveLength

### Beschikbaarheid

Flash MX 2004.

### Gebruik

stroke.waveLength

### Beschrijving

Eigenschap; een tekenreeks die de golflengte van een ongelijke lijn aangeeft. Deze eigenschap is alleen beschikbaar wanneer de eigenschap `stroke.style` is ingesteld op `ragged` (zie [stroke.style](#)). Acceptabele waarden zijn "very short", "short", "medium" en "long".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `waveLength` voor de streekstijl `ragged` ingesteld op `short`:

```
var myStroke = fl.getDocumentDOM().getCustomStroke();  
myStroke.style = "ragged";  
myStroke.pattern = "random";  
myStroke.waveHeight = 'flat';  
myStroke.waveLength = "short";  
fl.getDocumentDOM().setCustomStroke(myStroke);
```



# Hoofdstuk 39: Object swfPanel

## Beschikbaarheid

Flash CS4 Professional.

## Beschrijving

Het object `swfPanel` staat voor een Window SWF-deelvenster. Dit zijn SWF-bestanden die toepassingen implementeren die u vanuit de ontwerpomgeving van Flash kunt uitvoeren. Deze deelvensters zijn beschikbaar via het menu Venster > Andere deelvensters. Window SWF-deelvensters worden standaard opgeslagen in een submap van de map Configuration (zie “[JSFL-bestanden opslaan](#)” op pagina 2). In Windows XP vindt u de map bijvoorbeeld in *opstartschijf:\Documents and Settings\gebruiker\Local Settings\Application Data\Adobe\Flash CS4\taal\Configuration\WindowSWF*. Een voorbeeld van een Window SWF-deelvenster kunt u vinden in “[Voorbeeld van deelvenster Bitmap overtrekken](#)” op pagina 14. De array met geregistreerde Window SWF-deelvensters is opgeslagen in de eigenschap `fl.swfPanels`.

## Overzicht van methoden

U kunt de volgende methode gebruiken met het object `swfPanel`:

Methode	Beschrijving
<code>swfPanel.call()</code>	Deze methode werkt in combinatie met de methoden <code>ExternalInterface.addCallback()</code> en <code>MMEExecute()</code> van ActionScript om te communiceren met het SWF-deelvenster vanuit de ontwerpomgeving.
“ <code>swfPanel.setFocus()</code> ” op pagina 422	Hiermee wordt de toetsenbordfocus op het opgegeven SWF-deelvenster geplaatst.

## Overzicht van eigenschappen

U kunt de volgende eigenschappen gebruiken met het object `swfPanel`:

Eigenschap	Beschrijving
<code>swfPanel.name</code>	Alleen-lezen; een tekenreeks met de naam van het opgegeven Window SWF-deelvenster.
<code>swfPanel.path</code>	Alleen-lezen; een tekenreeks met het pad naar het SWF-bestand dat wordt gebruikt in het opgegeven Window SWF-deelvenster.

## swfPanel.call()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
swfPanel.call(request)
```

### Parameters

`request` Parameters die worden doorgegeven aan de functie (zie “Beschrijving” en “Voorbeeld” hieronder).

### Geretourneerde waarde

`null` of een tekenreeks die wordt geretourneerd door de aanroep van de functie. Het resultaat van de functie kan een lege tekenreeks zijn.

### Beschrijving

Methode; werkt in combinatie met de methoden `ExternalInterface.addCallback()` en `MMExecute()` van ActionScript om te communiceren met het SWF-deelvenster vanuit de ontwerpomgeving.

### Voorbeeld

In het volgende voorbeeld ziet u hoe u ActionScript- en JavaScript-code gebruikt om een Window SWF-deelvenster te maken en ermee communiceert vanuit de ontwerpomgeving.

- 1 Maak een ActionScript 3.0 FLA-bestand, stel de kleur in op middengrijs en stel het formaat in op 400 pixels bij 250 pixels.
- 2 Plaats een dynamisch tekstvak in het midden van het werkgebied, stel de instantienaam in op `myTextField` en typ het woord "Status" in het tekstvak.
- 3 Stel de overige tekstvakeigenschappen ongeveer als volgt in:

- Gecentreerd
- 355 pixels breed bij 46 pixels hoog
- Lettertype Times New Roman, 28 punten, rood

- 4 Voeg de volgende ActionScript-code toe:

```
// Here's the callback function to be called from JSAPI
function callMeFromJavascript(arg:String):void
{
    try {
        var name:String = String(arg);
        myTextField.text = name;
    } catch (e:Error) {
    }
}

// Expose the callback function as "callMySWF"
ExternalInterface.addCallback("callMySWF", callMeFromJavascript);

// run the JSAPI to wire up the callback
MMExecute("fl.runScript( fl.configURI + \"WindowSWF/fileOp.jsfl\" );");

MMExecute("fl.trace(\"AS3 File Status Panel Initialized\");");
```

- 5 Sla het bestand op als `fileStatus.fla` en publiceer het SWF-bestand met de standaardpublicatie-instellingen.
- 6 Sluit Flash.
- 7 Kopieer het bestand `fileStatus.swf` naar de map `WindowSWF`, een submap van de map `Configuration` (zie “[JSFL-bestanden opslaan](#)” op pagina 2). In Windows XP vindt u de map bijvoorbeeld in `opstartschijf:\Documents and Settings\gebruiker\Local Settings\Application Data\Adobe\Flash CS4\taal\Configuration\WindowSWF`.
- 8 Start Flash.
- 9 Maak een JSFL-bestand met de volgende code:

**Object swfPanel**

```

function callMyPanel(panelName, arg)
{
    if(fl.swfPanels.length > 0){
        for(x = 0; x < fl.swfPanels.length; x++){
            // look for a SWF panel of the specified name, then call the specified AS3
            function
            // in this example, the panel is named "test" and the AS3 callback is "callMySWF"
            if(fl.swfPanels[x].name == panelName) // name busted?
            {
                fl.swfPanels[x].call("callMySWF", arg);
                break;
            }
        }
    }
    else
        fl.trace("no panels");
}

// define the various handlers for events
documentClosedHandler = function () { callMyPanel("fileStatus", "Document Closed");};
fl.addEventListener("documentClosed", documentClosedHandler );

var dater = "New Document";
documentNewHandler = function () { callMyPanel("fileStatus", dater );};
fl.addEventListener("documentNew", documentNewHandler );

documentOpenedHandler = function () { callMyPanel("fileStatus", "Document Opened");};
fl.addEventListener("documentOpened", documentOpenedHandler );

```

10 Sla het JSFL-bestand in dezelfde map op als het SWF-bestand, met de naam fileOp.jsfl.

11 Kies Venster > Andere deelvvensters > fileStatus.

Nu wordt bij het maken, openen en sluiten van FLA-bestanden in het deelvvenster fileStatus vermeld welke actie u hebt uitgevoerd.

## swfPanel.name

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

swfPanel.name

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die de naam van het opgegeven Windows SWF-deelvvenster vertegenwoordigt.

### Voorbeeld

Met de volgende code wordt de naam van het eerste geregistreerde Window SWF-deelvvenster weergegeven in het deelvvenster Uitvoer:

```
fl.trace(fl.swfPanels[0].name);
```

#### Zie ook

[swfPanel.path](#), [fl.swfPanels](#)

## swfPanel.path

#### Beschikbaarheid

Flash CS4 Professional.

#### Gebruik

```
swfPanel.path
```

#### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die het pad vertegenwoordigt naar het SWF-bestand dat wordt gebruikt in het opgegeven Windows SWF-deelvenster.

#### Voorbeeld

Met de volgende code wordt het pad van het SWF-bestand dat wordt gebruikt in het eerste geregistreerde Window SWF-deelvenster, weergegeven in het deelvenster Uitvoer:

```
fl.trace(fl.swfPanels[0].path);
```

#### Zie ook

[swfPanel.name](#), [fl.swfPanels](#)

## swfPanel.setFocus()

#### Beschikbaarheid

Flash CS5.5 Professional.

#### Gebruik

```
swfPanel.setFocus()
```

#### Beschrijving

Methode: hiermee wordt de toetsenbordfocus op het opgegeven SWF-deelvenster geplaatst.

#### Voorbeeld

Met de volgende code wordt de focus op het SWF-deelvenster Project geplaatst:

Voer de volgende stappen uit voordat u deze opdracht uitvoert:

- 1 Koppel het deelvenster Project los, zodat het een zwevend deelvenster wordt.
- 2 Open het dialoogvenster Bestand maken via het deelvenster Project en klik vervolgens op het werkgebied.
- 3 Druk een aantal keer op de toets Tab om er zeker van te zijn dat de focus niet is gericht op het deelvenster Project.

- 4 Voer het onderstaande script uit vanaf het menu Opdrachten. Plaats hiertoe een JSFL-bestand met de onderstaande code in de directory `user/config/Commands`:
- 5 Druk op de toets Tab. Er wordt een cursor met invoegpositie weergegeven in een van de tekstvelden van het dialoogvenster Bestand maken.

```
flash.getSwfPanel("Project").setFocus();
```

**Zie ook**

[swfPanel.name](#), [fl.swfPanels](#)

# Hoofdstuk 40: Object SymbolInstance

**Overerving** [Object Element](#) > [Object Instance](#) > Object SymbolInstance

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

SymbolInstance is een subklasse van het object Instance en vertegenwoordigt een symbool in een frame (zie [Object Instance](#)).

## Overzicht van eigenschappen

Het object SymbolInstance heeft naast de eigenschappen van het object Instance ook de volgende eigenschappen:

Eigenschap	Beschrijving
<code>symbolInstance.accName</code>	Een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid.
<code>symbolInstance.actionScript</code>	Een tekenreeks die de aan het symbool toegekende handelingen aangeeft.
<code>symbolInstance.backgroundColor</code>	Een tekenreeks die de mattekleur bepaalt wanneer de optie Dekkend is geselecteerd.
<code>symbolInstance.bitmapRenderMode</code>	Een tekenreeks die het weergavetype voor een symboolinstantie bepaalt.
<code>symbolInstance.blendMode</code>	Een tekenreeks die de overvloeimodus voor een filmclipsymbool aangeeft.
<code>symbolInstance.buttonTracking</code>	Een tekenreeks die alleen voor knopsymbolen dezelfde eigenschap instelt als het pop-upmenu voor Bijhouden als knop of Bijhouden als menu-item in de Eigenschapcontrole.
<code>symbolInstance.cacheAsBitmap</code>	Een Booleaanse waarde die aangeeft of bitmap in cache plaatsen tijdens runtime is ingeschakeld.
<code>symbolInstance.colorAlphaAmount</code>	Een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie en dat de geavanceerde effect-alpha-instellingen aangeeft; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole om de besturingselementen rechts van het dialoogvenster aan te passen.
<code>symbolInstance.colorAlphaPercent</code>	Een geheel getal dat een deel van de kleurtransformatie voor de instantie bepaalt; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (besturingselementen voor percentages links van het dialoogvenster).
<code>symbolInstance.colorBlueAmount</code>	Een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties.
<code>symbolInstance.colorBluePercent</code>	Een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (besturingselementen voor percentages links van het dialoogvenster).
<code>symbolInstance.colorGreenAmount</code>	Een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties. Acceptabele waarden liggen tussen -255 en 255.

Eigenschap	Beschrijving
<code>symbolInstance.colorGreenPercent</code>	Onderdeel van de kleurtransformatie voor de instantie; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (besturingselementen voor percentages links van het dialoogvenster).
<code>symbolInstance.colorMode</code>	Een tekenreeks die de kleurmodus aangeeft, zoals wordt geïdentificeerd in het pop-up menu Kleur van Eigenschapcontrole voor symbolen.
<code>symbolInstance.colorRedAmount</code>	Een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie, is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties.
<code>symbolInstance.colorRedPercent</code>	Onderdeel van de kleurtransformatie voor de instantie; is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (besturingselementen voor percentages links van het dialoogvenster).
<code>symbolInstance.description</code>	Een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid.
<code>symbolInstance.filters</code>	Een array met objecten Filter (zie <a href="#">Object Filter</a> ).
<code>symbolInstance.firstFrame</code>	Een op nul gebaseerd geheel getal dat het eerste frame aangeeft dat in de tijdlijn van de afbeelding moet verschijnen.
<code>symbolInstance.forceSimple</code>	Een Booleaanse waarde die de toegankelijkheid van de onderliggende item van het object in- en uitschakelt; is gelijk aan de omgekeerde logica van de instelling Onderliggende objecten toegankelijk maken in het deelvenster Toegankelijkheid.
<code>symbolInstance.loop</code>	Een tekenreeks die voor grafische symbolen dezelfde eigenschap instelt als het pop-up menu Lus in de Eigenschapcontrole.
<code>symbolInstance.shortcut</code>	Een tekenreeks die gelijk is aan de aan het symbool toegewezen sneltoets; is gelijk aan het veld Sneltoets in het deelvenster Toegankelijkheid.
<code>symbolInstance.silent</code>	Een Booleaanse waarde die de toegankelijkheid van het object in- en uitschakelt; gelijk aan de omgekeerde logica van de instelling Object toegankelijk maken in het deelvenster Toegankelijkheid.
<code>symbolInstance.symbolType</code>	Een tekenreeks die het type symbool aangeeft; gelijk aan de waarde voor Gedrag in het dialoogvenster Nieuw symbool maken en Omzetten in symbool.
<code>symbolInstance.tabIndex</code>	Een geheel getal dat gelijk is aan het veld Tabindex in het deelvenster Toegankelijkheid.
<code>"symbolInstance.usesBackgroundColor"</code> op pagina 436	Een booleaanse waarde die de weergave-indeling bepaalt.
<code>"symbolInstance.visible"</code> op pagina 437	Een booleaanse waarde die opgeeft of de instantie al dan niet zichtbaar is.

## symbolInstance.accName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.accName`

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid. Schermlezers identificeren objecten door de naam hardop voor te lezen. Deze eigenschap is niet beschikbaar voor afbeeldingssymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de waarde voor de naam van het deelvenster Toegankelijkheid van het object opgeslagen in de variabele `theName`:

```
var theName = fl.getDocumentDOM().selection[0].accName;
```

In het volgende voorbeeld wordt de waarde voor de naam van het deelvenster Toegankelijkheid van het object ingesteld op `Home Button`:

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

## symbolInstance.actionScript

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.actionScript
```

### Beschrijving

Eigenschap; een tekenreeks die de aan het symbool toegekende handelingen aangeeft. Is alleen van toepassing op filmclips en knopinstanties. Voor een afbeeldingssymbolinstantie wordt de waarde `undefined` geretourneerd.

### Voorbeeld

In het volgende voorbeeld wordt een handeling `onClipEvent` toegekend aan het eerste item in het eerste frame van de eerste laag in de tijdlijn:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].actionScript  
= "onClipEvent(enterFrame) {trace('movie clip enterFrame');}";
```

## symbolInstance.backgroundColor

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
symbolInstance.backgroundColor
```

### Beschrijving

Eigenschap; een tekenreeks die de mattekleur bepaalt wanneer de 24-bits modus is geselecteerd voor de instantie. Dit is een tekenreeks in hexadecimale `#rrggbb`-indeling of een geheel getal dat de waarde bevat.



### Voorbeeld

Het volgende voorbeeld wijst de achtergrondkleur zwart toe aan de symboolinstantie:

```
var bitmapInstance = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];  
bitmapInstance.backgroundColor = "#000000";
```

## symbolInstance.bitmapRenderMode

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
symbolInstance.bitmapRenderMode
```

### Beschrijving

Eigenschap; een tekenreeks die het weergavetype van het symbool instelt.

Tot de acceptabele waarden behoren:

- "none"
- "cache" - hiermee wordt het symbool zodanig ingesteld dat het tijdens de runtime door Flash Player als een bitmap in de cache wordt opgeslagen.
- "export" - hiermee stelt u in dat het symbool als een bitmap wordt geëxporteerd wanneer het SWF-bestand wordt gecompileerd.

De oudere eigenschap "[symbolInstance.cacheAsBitmap](#)" op pagina 428 is vergelijkbaar met deze eigenschap, maar biedt minder keuzemogelijkheden omdat het een Booleaanse eigenschap is. In de toekomst wordt de eigenschap `cacheAsBitmap` mogelijk gedeactiveerd en gebruikers moeten overstappen op deze nieuwe eigenschap. De opties `true/false` in de Booleaanse eigenschap `cacheAsBitmap` zijn dezelfde als de waarden 'cache'/none' voor deze nieuwe eigenschap.

### Voorbeeld

Het volgende voorbeeld wijst "export" toe aan `bitmapRenderMode` van het symbool:

```
var symbol = fl.getDocumentDOM().selection[0];  
fl.trace(symbol.bitmapRenderMode);  
symbol.bitmapRenderMode = "export";
```

## symbolInstance.blendMode

### Beschikbaarheid

Flash 8.

### Gebruik

```
symbolInstance.blendMode
```

### Beschrijving

Eigenschap; een tekenreeks die de overvloeimodus voor een filmclipsymbool aangeeft. Acceptabele waarden zijn "normal", "layer", "multiply", "screen", "overlay", "hardlight", "lighten", "darken", "difference", "add", "subtract", "invert", "alpha" en "erase".

### Voorbeeld

In het volgende voorbeeld wordt de overvloeimodus voor het eerste filmclipsymbool in het eerste frame op het eerste niveau ingesteld op add:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].blendMode = "add";
```

### Zie ook

[document.setBlendMode\(\)](#)

## symbolInstance.buttonTracking

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.buttonTracking`

### Beschrijving

Eigenschap; een tekenreeks die, alleen voor knopsymbolen, dezelfde eigenschap instelt als het pop-upmenu voor Bijhouden als knop of Bijhouden als menu-item in de Eigenschapcontrole. Voor andere typen symbolen wordt deze eigenschap genegeerd. Acceptabele waarden zijn "button" en "menu".

### Voorbeeld

In het volgende voorbeeld wordt het eerste symbool in het eerste frame van de eerste laag op de tijdlijn ingesteld op Bijhouden als menu-item, op voorwaarde dat het symbool een knop is:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].buttonTracking = "menu";
```

## symbolInstance.cacheAsBitmap

### Beschikbaarheid

Flash 8.

### Gebruik

`symbolInstance.cacheAsBitmap`

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of bitmap in cache plaatsen tijdens runtime is ingeschakeld.

**Opmerking:** Vanaf Flash Professional CS5.5 moeten gebruikers in plaats van deze eigenschap de eigenschap "[symbolInstance.bitmapRenderMode](#)" op pagina 427 gebruiken.

### Voorbeeld

In het volgende voorbeeld wordt bitmap in cache plaatsen tijdens runtime voor het eerste element in het eerste frame op de eerste laag geactiveerd:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].cacheAsBitmap = true;
```

## symbolInstance.colorAlphaAmount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.colorAlphaAmount
```

### Beschrijving

Eigenschap; een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie; hiermee worden de geavanceerde effect-alpha-instellingen aangegeven. Deze eigenschap is is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole om de besturingselementen rechts van het dialoogvenster aan te passen. Deze waarde vermindert of vermeerderd de tint- en alfa-waarden met een constante hoeveelheid. Deze waarde wordt toegevoegd aan de huidige waarde. Deze eigenschap is zeer nuttig in combinatie met [symbolInstance.colorAlphaPercent](#). Acceptabele waarden liggen tussen -255 en 255.

### Voorbeeld

In het volgende voorbeeld wordt de alfa-instelling van de geselecteerde symboolinstantie verminderd met 100:

```
fl.getDocumentDOM().selection[0].colorAlphaAmount = -100;
```

## symbolInstance.colorAlphaPercent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.colorAlphaPercent
```

### Beschrijving

Eigenschap; een geheel getal dat een deel van de kleurtransformatie voor de instantie bepaalt. Deze eigenschap is is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (de besturingselementen voor percentages links van het dialoogvenster). Deze waarde wijzigt de tint- en alfa-waarden in een bepaald percentage. Acceptabele waarden liggen tussen -100 en 100. Zie ook [symbolInstance.colorAlphaAmount](#).

### Voorbeeld

In het volgende voorbeeld wordt het `colorAlphaPercent` van de geselecteerde symboolinstantie ingesteld op 80:

```
fl.getDocumentDOM().selection[0].colorAlphaPercent = 80;
```

## symbolInstance.colorBlueAmount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.colorBlueAmount`

### Beschrijving

Eigenschap; een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie. Deze eigenschap is is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties. Acceptabele waarden liggen tussen -255 en 255.

## symbolInstance.colorBluePercent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.colorBluePercent`

### Beschrijving

Eigenschap; een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie. Deze eigenschap is is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (de besturingselementen voor percentages links van het dialoogvenster). Deze waarde stelt de blauwe waarden in op een bepaald percentage. Acceptabele waarden liggen tussen -100 en 100.

### Voorbeeld

In het volgende voorbeeld wordt `colorBluePercent` van de geselecteerde symboolinstantie ingesteld op 80:

```
fl.getDocumentDOM().selection[0].colorBluePercent = 80;
```

## symbolInstance.colorGreenAmount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.colorGreenAmount`

### Beschrijving

Eigenschap; een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie. Deze eigenschap is is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties. Acceptabele waarden liggen tussen -255 en 255.

## symbolInstance.colorGreenPercent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.colorGreenPercent
```

### Beschrijving

Eigenschap; onderdeel van de kleurtransformatie voor de instantie. Deze eigenschap is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (de besturingselementen voor percentages links van het dialoogvenster). Deze waarde stelt de groene waarden in op een bepaald percentage. Acceptabele waarden liggen tussen -100 en 100.

### Voorbeeld

In het volgende voorbeeld wordt `colorGreenPercent` van de geselecteerde symboolinstantie ingesteld op 70:

```
fl.getDocumentDOM().selection[0].colorGreenPercent = 70;
```

## symbolInstance.colorMode

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.colorMode
```

### Beschrijving

Eigenschap; een tekenreeks die de kleurmodus aangeeft, zoals wordt geïdentificeerd in het pop-up menu Kleur van Eigenschapcontrole voor symbolen. Acceptabele waarden zijn "none", "brightness", "tint", "alpha" en "advanced".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `colorMode` van het eerste element in het eerste frame van de eerste laag op de tijdlijn gewijzigd in alpha:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].colorMode = "alpha";
```

## symbolInstance.colorRedAmount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.colorRedAmount
```

### Beschrijving

Eigenschap; een geheel getal dat onderdeel is van de kleurtransformatie voor de instantie. Deze eigenschap is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties. Acceptabele waarden liggen tussen -255 en 255.

### Voorbeeld

In het volgende voorbeeld wordt `colorRedAmount` van de geselecteerde symboolinstantie ingesteld op 255:

```
fl.getDocumentDOM().selection[0].colorRedAmount = 255;
```

## symbolInstance.colorRedPercent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.colorRedPercent
```

### Beschrijving

Eigenschap; onderdeel van de kleurtransformatie voor de instantie. Deze eigenschap is gelijk aan de instelling Kleur > Geavanceerd in de Eigenschapcontrole voor instanties (de besturingselementen voor percentages links van het dialoogvenster). Deze waarde stelt de rode waarden in op een bepaald percentage. Acceptabele waarden liggen tussen -100 en 100.

### Voorbeeld

In het volgende voorbeeld wordt `colorRedPercent` van de geselecteerde symboolinstantie ingesteld op 10:

```
fl.getDocumentDOM().selection[0].colorRedPercent = 10;
```

## symbolInstance.description

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.description
```

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid. De beschrijving wordt gelezen door de schermlezer. Deze eigenschap is niet beschikbaar voor afbeeldingssymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de waarde voor de beschrijving van het deelvenster Toegankelijkheid van het object opgeslagen in de variabele `theDescription`:

```
var theDescription = fl.getDocumentDOM().selection[0].description;
```

In het volgende voorbeeld wordt de waarde voor de beschrijving van het deelvenster Toegankelijkheid ingesteld op Click the home button to go to home:

```
fl.getDocumentDOM().selection[0].description= "Click the home button to go to home";
```

## symbolInstance.filters

### Beschikbaarheid

Flash 8.

### Gebruik

```
symbolInstance.filters
```

### Beschrijving

Eigenschap; een array van objecten Filter (zie [Object Filter](#)). Als u filtereigenschappen wilt wijzigen, schrijft u niet rechtstreeks naar deze array. In plaats daarvan haalt u de array op, stelt u de individuele eigenschappen in en stelt u de array vervolgens in met de nieuwe eigenschappen.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de filter op index 0 getraceerd. Als het een gloedfilter is, wordt de eigenschap `blurX` ingesteld op 100 en wordt de nieuwe waarde geschreven naar de filterarray.

```
var filterName =  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters[0].name;  
fl.trace(filterName);  
var filterArray = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters;  
if (filterName == 'glowFilter'){  
    filterArray[0].blurX = 100;  
}  
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters = filterArray;
```

## symbolInstance.firstFrame

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.firstFrame
```

### Beschrijving

Eigenschap; een op nul gebaseerd geheel getal dat het eerste frame aangeeft dat in de tijdlijn van de afbeelding moet verschijnen. Deze eigenschap wordt alleen toegepast op grafische symbolen. Hiermee wordt dezelfde eigenschap ingesteld als met het veld Eerste in de Eigenschapcontrole. Voor andere type symbolen is deze eigenschap `undefined`.

### Voorbeeld

In het volgende voorbeeld wordt vastgelegd dat Frame 10 als eerste frame in de tijdlijn van het opgegeven element moet verschijnen:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].firstFrame = 10;
```

## symbolInstance.forceSimple

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.forceSimple
```

### Beschrijving

Eigenschap; een Booleaanse waarde die de toegankelijkheid van de onderliggende items van het object in- en uitschakelt. Deze eigenschap is gelijk aan de omgekeerde logica van de instelling Onderliggende objecten toegankelijk maken in het deelvenster Toegankelijkheid. Dit betekent bijvoorbeeld dat de instelling `forceSimple` is `true`, hetzelfde is als het uitschakelen van de optie Onderliggend object toegankelijk maken. `forceSimple` is `false`, is hetzelfde als het inschakelen van de optie Onderliggend object toegankelijk maken.

Deze eigenschap is alleen voor objecten MovieClip beschikbaar.

### Voorbeeld

In het volgende voorbeeld wordt gecontroleerd of de onderliggende items van het object toegankelijk zijn; de retourwaarde `false` betekent dat de items toegankelijk zijn:

```
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;
```

In het volgende voorbeeld wordt ingesteld dat de onderliggende items van het object toegankelijk zijn:

```
fl.getDocumentDOM().selection[0].forceSimple = false;
```

## symbolInstance.loop

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.loop
```

### Beschrijving

Eigenschap; een tekenreeks die voor grafische symbolen dezelfde eigenschap instelt als het pop-up menu Lus in de Eigenschapcontrole. Voor andere type symbolen is deze eigenschap `undefined`. Acceptabele waarden zijn "loop", "play once" en "single frame", waarmee de animatie van de afbeelding kan worden ingesteld.

### Voorbeeld

In het volgende voorbeeld wordt het eerste symbool in het eerste frame van de eerste laag op de tijdlijn ingesteld op `single frame` (één bepaald frame van de grafische tijdlijn weergeven), op voorwaarde dat het symbool een afbeelding is:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].loop = 'single frame';
```



## symbolInstance.shortcut

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.shortcut`

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan de sneltoets die is toegewezen aan het symbool. Deze eigenschap is gelijk aan het veld Sneltoets in het deelvenster Toegankelijkheid. De toets wordt gelezen door schermlezers. Deze eigenschap is niet beschikbaar voor afbeeldingssymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de waarde voor de sneltoets van het object opgeslagen in de variabele `theShortcut`:

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
```

In het volgende voorbeeld wordt de sneltoets van het object ingesteld op `Ctrl+i`:

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

## symbolInstance.silent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`symbolInstance.silent`

### Beschrijving

Eigenschap; een Booleaanse waarde die de toegankelijkheid van het object in- of uitschakelt. Deze eigenschap is gelijk aan de omgekeerde logica van de instelling Object toegankelijk maken in het deelvenster Toegankelijkheid. Dit betekent bijvoorbeeld dat de instelling `silent` is `true`, hetzelfde is als het uitschakelen van de optie Object toegankelijk maken. `silent` is `false`, is hetzelfde als het inschakelen van de optie Object toegankelijk maken.

Deze eigenschap is niet beschikbaar voor grafische objecten.

### Voorbeeld

In het volgende voorbeeld wordt gecontroleerd of het object toegankelijk is; de retourwaarde `false` betekent dat dit het geval is:

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

In het volgende voorbeeld wordt het object toegankelijk gemaakt:

```
fl.getDocumentDOM().selection[0].silent = false;
```

## symbolInstance.symbolType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.symbolType
```

### Beschrijving

Eigenschap; een tekenreeks die het type symbool opgeeft. Deze eigenschap is gelijk aan de waarde voor Gedrag in het dialoogvenster Nieuw symbool maken en Omzetten in symbool. Acceptabele waarden zijn "button", "movie clip" en "graphic".

### Voorbeeld

In het volgende voorbeeld wordt ingesteld dat het eerste symbool in het eerste frame van de eerste laag in de tijdlijn van het huidige document zich gedraagt als grafisch symbool:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].symbolType = "graphic";
```

## symbolInstance.tabIndex

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolInstance.tabIndex
```

### Beschrijving

Eigenschap; een geheel getal dat gelijk is aan het veld Tabindex in het deelvenster Toegankelijkheid. Hiermee wordt een tabvolgorde gemaakt waarin objecten worden benaderd wanneer de gebruiker op de Tab-toets drukt. Deze eigenschap is niet beschikbaar voor afbeeldingssymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `tabIndex` van het object `mySymbol` ingesteld op 3 en verschijnt die waarde in het deelvenster Uitvoer:

```
var mySymbol = fl.getDocumentDOM().selection[0];  
mySymbol.tabIndex = 3;  
fl.trace(mySymbol.tabIndex);
```

## symbolInstance.usesBackgroundColor

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

`symbolInstance.usesBackgroundColor`

### Beschrijving

Eigenschap; een booleaanse waarde die aangeeft of de 24-bits modus (`true`) of de 32-bits modus (`false`) moet worden gebruikt voor de instantie. Indien `true`, wordt de voor de instantie opgegeven `backgroundColor` gebruikt.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `usesBackgroundColor` van een instantie ingesteld op `true`:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].useTransparentBackground = true;
```

## **symbolInstance.visible**

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

`symbolInstance.visible`

### Beschrijving

Eigenschap; een booleaanse waarde die de eigenschap `Visible` van een object instelt op aan (`true`) of uit (`false`).

### Voorbeeld

In het volgende voorbeeld wordt de zichtbaarheid van het eerste item in het eerste frame van de eerste laag ingesteld op `false`:

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].visible = false;
```

# Hoofdstuk 41: Object SymbolItem

**Overerving** [Object Item](#) > Object SymbolItem

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object SymbolItem is een subklasse van het [Object Item](#).

## Overzicht van methoden

Naast de methoden van het object Item kunt u de volgende methoden gebruiken voor het object SymbolItem:

Methode	Beschrijving
<code>symbolItem.convertToCompiledClip()</code>	Hiermee wordt een symbol-item in de bibliotheek geconverteerd naar een gecompileerde filmclip.
<code>symbolItem.exportSWC()</code>	Hiermee wordt het symbol-item geëxporteerd naar een SWC-bestand.
<code>symbolItem.exportSWF()</code>	Hiermee wordt het symbol-item geëxporteerd naar een SWF-bestand.

## Overzicht van eigenschappen

Naast de eigenschappen van het object Item zijn de volgende eigenschappen beschikbaar voor het object SymbolItem:

Eigenschap	Beschrijving
<code>symbolItem.scalingGrid</code>	Een Booleaanse waarde waarmee wordt opgegeven of 9-delige schaling is ingeschakeld voor het item.
<code>symbolItem.scalingGridRect</code>	Een object Rectangle dat de locaties van de vier hulplijnen voor 9-delige schaling aangeeft.
<code>symbolItem.sourceAutoUpdate</code>	Een Booleaanse waarde die opgeeft of het item wordt bijgewerkt wanneer het FLA-bestand wordt gepubliceerd.
<code>symbolItem.sourceFilePath</code>	Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad voor het FLA-bronbestand opgeeft.
<code>symbolItem.sourceLibraryName</code>	Een tekenreeks die de naam opgeeft van het item in de bibliotheek met bronbestanden.
<code>symbolItem.symbolType</code>	Een tekenreeks die het type symbool opgeeft.
<code>symbolItem.timeline</code>	Alleen-lezen; een <a href="#">Object Timeline</a> .

## `symbolItem.convertToCompiledClip()`

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.convertToCompiledClip()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een symbol-item in de bibliotheek geconverteerd naar een gecompileerde filmclip.

### Voorbeeld

In het volgende voorbeeld wordt een item in de bibliotheek geconverteerd naar een gecompileerde filmclip:

```
fl.getDocumentDOM().library.items[3].convertToCompiledClip();
```

## symbolItem.exportSWC()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.exportSWC(outputURI)
```

### Parameters

**outputURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het SWC-bestand opgeeft waarnaar de methode het symbool exporteert. De *outputURI* moet verwijzen naar een lokaal bestand. Flash maakt geen map als *outputURI* niet bestaat.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het symbol-item geëxporteerd naar een SWC-bestand.

### Voorbeeld

In het volgende voorbeeld wordt een item in de bibliotheek geëxporteerd naar het SWC-bestand met de naam `mySymbol.swc` in de map `tests`:

```
fl.getDocumentDOM().library.selectItem("mySymbol");  
var currentSelection = fl.getDocumentDOM().library.getSelectedItems();  
currentSelection[0].exportSWC("file:///Macintosh HD/SWCDirectory/mySymbol.swc");
```

## symbolItem.exportSWF()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.exportSWF(outputURI)
```

### Parameters

**outputURI** Een tekenreeks, uitgedrukt als bestand:/// URI, die het SWF-bestand opgeeft waarnaar de methode het symbool exporteert. De *outputURI* moet verwijzen naar een lokaal bestand. Flash maakt geen map als *outputURI* niet bestaat.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het symbol-item geëxporteerd naar een SWF-bestand.

### Voorbeeld

In het volgende voorbeeld wordt een item in de bibliotheek geëxporteerd naar het bestand `my.swf` in de map `tests`:

```
fl.getDocumentDOM().library.items[0].exportSWF("file:///c:/tests/my.swf");
```

## symbolItem.scalingGrid

### Beschikbaarheid

Flash 8.

### Gebruik

```
symbolItem.scalingGrid
```

### Beschrijving

Eigenschap; een Booleaanse waarde waarmee wordt opgegeven of 9-delige schaling is ingeschakeld voor het item.

### Voorbeeld

In het volgende voorbeeld wordt 9-delige schaling ingeschakeld voor een item in de bibliotheek:

```
fl.getDocumentDOM().library.items[0].scalingGrid = true;
```

### Zie ook

[symbolItem.scalingGridRect](#)

## symbolItem.scalingGridRect

### Beschikbaarheid

Flash 8.

### Gebruik

```
symbolItem.scalingGridRect
```

### Beschrijving

Eigenschap; een object Rectangle dat de locaties van de vier hulplijnen voor 9-delige schaling aangeeft. Zie [document.addNewRectangle\(\)](#) voor informatie over de notatie van de rechthoek.

### Voorbeeld

In het volgende voorbeeld wordt de locatie van de 9-delige schaling opgegeven.

```
fl.getDocumentDOM().library.items[0].scalingGridRect = {left:338, top:237, right:3859,  
bottom:713};
```

### Zie ook

[symbolItem.scalingGrid](#)

## symbolItem.sourceAutoUpdate

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.sourceAutoUpdate
```

### Beschrijving

Eigenschap; een Booleaanse waarde die opgeeft of het item wordt bijgewerkt wanneer het FLA-bestand wordt gepubliceerd. De standaardwaarde is `false`. Wordt gebruikt voor gedeelde bibliotheeksymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `sourceAutoUpdate` ingesteld voor een bibliotheekitem:

```
fl.getDocumentDOM().library.items[0].sourceAutoUpdate = true;
```

## symbolItem.sourceFilePath

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.sourceFilePath
```

### Beschrijving

Eigenschap; een tekenreeks, uitgedrukt als bestand:/// URI, die het pad voor het FLA-bronbestand opgeeft. Het pad moet een absoluut pad zijn, niet een relatief pad. Deze eigenschap wordt gebruikt voor gedeelde bibliotheeksymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `sourceFilePath` getoond in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceFilePath);
```

## symbolItem.sourceLibraryName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.sourceLibraryName
```

### Beschrijving

Eigenschap; een tekenreeks die de naam opgeeft van het item in de bibliotheek met bronbestanden. Wordt gebruikt voor gedeelde bibliotheeksymbolen.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `sourceLibraryName` getoond in het deelvenster Uitvoer:

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceLibraryName);
```

## symbolItem.symbolType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
symbolItem.symbolType
```

### Beschrijving

Eigenschap; een tekenreeks die het type symbool opgeeft. Acceptabele waarden zijn "movie clip", "button" en "graphic".

### Voorbeeld

In het volgende voorbeeld wordt de huidige waarde van de eigenschap `symbolType` getoond, gewijzigd in `button` en vervolgens opnieuw weergegeven:

```
alert(fl.getDocumentDOM().library.items[0].symbolType);  
fl.getDocumentDOM().library.items[0].symbolType = "button";  
alert(fl.getDocumentDOM().library.items[0].symbolType);
```



## symbolItem.timeline

### Beschikbaarheid

Flash MX 2004.

### Gebruik

symbolItem.timeline

### Beschrijving

Alleen-lezen eigenschap; een [Object Timeline](#).

### Voorbeeld

In het volgende voorbeeld wordt het aantal lagen van de geselecteerde filmclip in de bibliotheek opgehaald en weergegeven:

```
var tl = fl.getDocumentDOM().library.getSelectedItems()[0].timeline;  
alert(tl.layerCount);
```

# Hoofdstuk 42: Object Text

**Overerving** [Object Element](#) > Object Text

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Text vertegenwoordigt één tekstitem in een document. Alle eigenschappen van de tekst hebben betrekking op het hele tekstblok.

Zie het overzicht met eigenschappen voor het [Object TextAttrs](#) voor het instellen van eigenschappen voor tekst in het tekstveld. Als u de eigenschappen wilt wijzigen van een selectie in een tekstveld, kunt u `document.setElementTextAttr()` gebruiken en een tekstbereik opgeven of de huidige selectie gebruiken.

Gebruik `document.setElementProperty()` om algemene eigenschappen van het geselecteerde tekstveld in te stellen. In het volgende voorbeeld wordt de `x`-waarde van het registratiepunt van het geselecteerde tekstveld ingesteld op 50:

```
fl.getDocumentDOM().setElementProperty("x", 50);
```

## Overzicht van methoden

Naast de methoden van het object Element, zijn de volgende methoden beschikbaar voor het object Text:

Methode	Beschrijving
<code>text.getTextAttr()</code>	Hiermee wordt het opgegeven kenmerk opgehaald voor de tekst die wordt bepaald door de optionele parameters <code>startIndex</code> en <code>endIndex</code> .
<code>text.getTextString()</code>	Hiermee wordt het opgegeven tekstbereik opgehaald.
<code>text.setTextAttr()</code>	Hiermee wordt het opgegeven kenmerk ingesteld dat is gekoppeld aan de tekst die wordt bepaald door <code>startIndex</code> en <code>endIndex</code> .
<code>text.setTextString()</code>	Hiermee wordt de tekst in dit object Text gewijzigd.

## Overzicht van eigenschappen

Naast de eigenschappen van het object Element, zijn de volgende eigenschappen beschikbaar voor het object Text:

Eigenschap	Beschrijving
<code>text.accName</code>	Een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid.
<code>text.antiAliasSharpness</code>	Een drijvende-kommawaarde waarmee de anti-aliasing scherpte van de tekst wordt opgegeven.
<code>text.antiAliasThickness</code>	Een drijvende-kommawaarde waarmee de anti-aliasing dikte van de tekst wordt opgegeven.
<code>text.autoExpand</code>	Een Booleaanse waarde die bepaalt of het kader voor statische tekstvelden breder mag worden of het kader voor dynamische tekst of invoertekst breder en hoger mag worden .
<code>text.border</code>	Een Booleaanse waarde die bepaalt of de rand om dynamische tekst of invoertekst wordt weergegeven ( <code>true</code> ) of verborgen ( <code>false</code> ).
<code>text.description</code>	Een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid.

Eigenschap	Beschrijving
<code>text.embeddedCharacters</code>	Een tekenreeks die de tekens bevat die moeten worden ingesloten. Dit is hetzelfde als het invoeren van tekst in het dialoogvenster Tekens insluiten.
<code>text.embedRanges</code>	Een tekenreeks die bestaat uit gehele getallen, gescheiden door scheidingstekens, die overeenkomen met de items die kunnen worden geselecteerd in het dialoogvenster Tekens insluiten.
<code>text.embedVariantGlyphs</code>	Een Booleaanse waarde die aangeeft of het insluiten van variantglyphs moet worden ingeschakeld.
<code>text.fontRenderingMode</code>	Een tekenreeks waarmee de rendermodus voor de tekst wordt opgegeven.
<code>text.length</code>	Alleen-lezen; een geheel getal dat het aantal tekens in het object Text aangeeft.
<code>text.lineType</code>	Een tekenreeks waarmee het regeltype wordt ingesteld op "single line", "multiline", "multiline no wrap" of "password".
<code>text.maxCharacters</code>	Een geheel getal dat het maximale aantal tekens aangeeft dat de gebruiker kan invoeren in dit object Text.
<code>text.orientation</code>	Een tekenreeks die de richting van het tekstveld aangeeft.
<code>text.renderAsHTML</code>	Een Booleaanse waarde waarmee wordt bepaald of de tekst wordt weergegeven als HTML en ingesloten HTML-codes worden geïnterpreteerd.
<code>text.scrollable</code>	Een Booleaanse waarde waarmee wordt bepaald of door de tekst kan worden geschoven ( <code>true</code> ) of niet ( <code>false</code> ).
<code>text.selectable</code>	Een Booleaanse waarde waarmee wordt bepaald of de tekst kan worden geselecteerd ( <code>true</code> ) of niet ( <code>false</code> ). Invoertekst kan altijd worden geselecteerd.
<code>text.selectionEnd</code>	Een op nul gebaseerd geheel getal dat de verschuiving aangeeft van het einde van een subselectie in een tekst.
<code>text.selectionStart</code>	Een op nul gebaseerd geheel getal dat de verschuiving aangeeft van het begin van een subselectie in een tekst.
<code>text.shortcut</code>	Een tekenreeks die gelijk is aan het veld Sneltoets in het deelvenster Toegankelijkheid.
<code>text.silent</code>	Een Booleaanse waarde die aangeeft of het object toegankelijk is.
<code>text.tabIndex</code>	Een geheel getal dat gelijk is aan het veld Tabindex in het deelvenster Toegankelijkheid.
<code>text.textRuns</code>	Alleen-lezen; een array met objecten TextRun.
<code>text.textType</code>	Een tekenreeks die het type van het tekstveld aangeeft. Acceptabele waarden zijn "static", "dynamic" en "input".
<code>text.useDeviceFonts</code>	Een Booleaanse waarde. Als de waarde <code>true</code> is, wordt tekst weergegeven met apparaatlettertypen.
<code>text.variableName</code>	Een tekenreeks die de inhoud van het object Text bevat.

## text.accName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.accName`

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Naam in het deelvenster Toegankelijkheid. Schermlezers identificeren objecten door de naam hardop voor te lezen. Deze eigenschap kan niet worden gebruikt met dynamische tekst.

### Voorbeeld

In het volgende voorbeeld wordt de naam van het object opgehaald:

```
var doc = fl.getDocumentDOM();  
var theName = doc.selection[0].accName;
```

In het volgende voorbeeld wordt de naam van het geselecteerde object ingesteld:

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

## text.antiAliasSharpness

### Beschikbaarheid

Flash 8.

### Gebruik

```
text.antiAliasSharpness
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde waarmee de anti-aliasing scherpte van de tekst wordt opgegeven. Deze eigenschap bepaalt hoe scherp de tekst wordt weergegeven, waarbij hogere waarden scherpere tekst aangeven. De waarde 0 geeft normale scherpte aan. Deze eigenschap is alleen beschikbaar als `text.fontRenderingMode` is ingesteld op `customThicknessSharpness`.

### Voorbeeld

Zie [text.fontRenderingMode](#).

### Zie ook

[text.antiAliasThickness](#), [text.fontRenderingMode](#)

## text.antiAliasThickness

### Beschikbaarheid

Flash 8.

### Gebruik

```
text.antiAliasThickness
```

### Beschrijving

Eigenschap; een drijvende-kommawaarde waarmee de anti-aliasing dikte van de tekst wordt opgegeven. Deze eigenschap bepaalt hoe dik de tekst wordt weergegeven, waarbij hogere waarden dikkere tekst aangeven. De waarde 0 geeft normale dikte aan. Deze eigenschap is alleen beschikbaar als `text.fontRenderingMode` is ingesteld op `customThicknessSharpness`.

### Voorbeeld

Zie `text.fontRenderingMode`.

### Zie ook

`text.antiAliasSharpness`, `text.fontRenderingMode`

## text.autoExpand

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.autoExpand`

### Beschrijving

Eigenschap; een Booleaanse waarde. Voor statische tekstvelden betekent de waarde `true` dat de breedte van het kader wordt aangepast zodat alle tekst wordt weergegeven. Voor dynamische velden of invoervelden betekent de waarde `true` dat de breedte en de hoogte van het kader worden aangepast zodat alle tekst wordt weergegeven.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `autoExpand` ingesteld op de waarde `true`:

```
fl.getDocumentDOM().selection[0].autoExpand = true;
```

## text.border

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.border`

### Beschrijving

Eigenschap; een Booleaanse waarde. Als de waarde `true` is, wordt een rand om de tekst weergegeven.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `border` ingesteld op de waarde `true`:

```
fl.getDocumentDOM().selection[0].border = true;
```

## text.description

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.description
```

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Beschrijving in het deelvenster Toegankelijkheid. De beschrijving wordt gelezen door de schermlezer.

### Voorbeeld

In het volgende voorbeeld wordt de beschrijving van het object opgehaald:

```
var doc = fl.getDocumentDOM();  
var desc = doc.selection[0].description;
```

In het volgende voorbeeld wordt de beschrijving van het object ingesteld:

```
var doc = fl.getDocumentDOM();  
doc.selection[0].description= "Enter your name here";
```

## text.embeddedCharacters

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.embeddedCharacters
```

### Beschrijving

Eigenschap; een tekenreeks die de tekens bevat die moeten worden ingesloten. Dit is hetzelfde als het invoeren van tekst in het dialoogvenster Tekens insluiten.

Deze eigenschap werkt alleen met dynamische tekst of invoertekst; bij andere teksttypen wordt een waarschuwing gegenereerd.

**Opmerking:** Vanaf Flash Professional CS5 wordt het insluiten van lettertypen beheerd op documentniveau in plaats van op tekstobjectniveau. Gebruik de eigenschap “[fontItem.embeddedCharacters](#)” op pagina 288 in plaats van de eigenschap `text.embeddedCharacters`.

### Voorbeeld

In het volgende voorbeeld wordt ervan uitgegaan dat het eerste of enige item in de huidige selectie een klassiek tekstobject is en wordt de eigenschap `embeddedCharacters` ingesteld op `abc`:

```
fl.getDocumentDOM().selection[0].embeddedCharacters = "abc";
```

## text.embedRanges

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.embedRanges
```

### Beschrijving

Eigenschap; een tekenreeks die bestaat uit gehele getallen, gescheiden door scheidingstekens, die overeenkomen met de items die kunnen worden geselecteerd in het dialoogvenster Tekens insluiten. Deze eigenschap werkt alleen met dynamische tekst of invoertekst; de eigenschap wordt genegeerd bij statische tekst.

Deze eigenschap komt overeen met het XML-bestand in de map Configuration/Font Embedding.

**Opmerking:** Vanaf Flash Professional CS5 wordt het insluiten van lettertypen beheerd op documentniveau in plaats van op tekstobjectniveau. Gebruik de eigenschap “[fontItem.embedRanges](#)” op pagina 289 in plaats van de eigenschap `text.embedRanges`.

### Voorbeeld

In het volgende voorbeeld wordt ervan uitgegaan dat het eerste of enige item in de huidige selectie een klassiek tekstobject is en wordt de eigenschap `embedRanges` ingesteld op "1|3|7":

```
var doc = fl.getDocumentDOM();  
doc.selection[0].embedRanges = "1|3|7";
```

In het volgende voorbeeld wordt de standaardwaarde van de eigenschap hersteld:

```
var doc = fl.getDocumentDOM();  
doc.selection[0].embedRanges = "";
```

## text.embedVariantGlyphs

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
text.embedVariantGlyphs
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of insluiting van variantglyphs moet worden ingeschakeld (`true`) of niet (`false`). Deze eigenschap werkt alleen met dynamische tekst of invoertekst; de eigenschap wordt genegeerd bij statische tekst. De standaardwaarde is `false`.

**Opmerking:** Vanaf Flash Professional CS5 wordt het insluiten van lettertypen beheerd op documentniveau in plaats van op tekstobjectniveau. Gebruik de eigenschap “[fontItem.embedVariantGlyphs](#)” op pagina 289 in plaats van de eigenschap `text.embedVariantGlyphs`. In Flash Professional CS5 heeft de eigenschap `text.embedVariantGlyphs` geen effect meer, omdat Flash altijd variantglyphs insluit voor TLF-tekst en nooit voor klassieke tekst.

### Voorbeeld

In het volgende voorbeeld wordt het insluiten van variantglyphs in het geselecteerde object Text ingeschakeld:

```
fl.getDocumentDOM().selection[0].embedVariantGlyphs = true;
```

### Zie ook

[fontItem.embedVariantGlyphs](#)

## text.fontRenderingMode

### Beschikbaarheid

Flash 8.

### Gebruik

`text.fontRenderingMode`

### Beschrijving

Eigenschap; een tekenreeks waarmee de rendermodus voor de tekst wordt opgegeven. Deze eigenschap bepaalt hoe de tekst wordt weergegeven in het werkgebied en in Flash Player. De volgende tabel bevat de acceptabele waarden:

Waarde van eigenschap	Rendermethode voor tekst
device	De tekst wordt gerenderd met apparaatlettertypen.
bitmap	Aliased tekst wordt gerenderd als een bitmap of zoals een pixellettertype.
standard	Tekst wordt gerenderd volgens de standaard anti-aliasing methode van Flash MX 2004. Dit is de beste instelling voor bewegende, zeer grote of scheefgetrokken tekst.
advanced	Tekst wordt gerenderd volgens de geavanceerde anti-aliasing technologie voor het renderen van lettertypen die in Flash 8 is geïmplementeerd. Hiermee worden de anti-aliasing en leesbaarheid verbeterd, vooral voor kleine tekst.
customThicknessSharpness	Hiermee kunt u aangepaste instellingen opgeven voor de scherpte en de dikte van de tekst bij het gebruik van de geavanceerde anti-aliasing technologie voor het renderen van lettertypen die in Flash 8 is geïmplementeerd.

### Voorbeeld

In het volgende voorbeeld ziet u hoe u met de waarde `customThicknessSharpness` de scherpte en de dikte van de tekst kunt opgeven:

```
fl.getDocumentDOM().setElementProperty("fontRenderingMode", "customThicknessSharpness");  
fl.getDocumentDOM().setElementProperty("antiAliasSharpness", 400);  
fl.getDocumentDOM().setElementProperty("antiAliasThickness", -200);
```

### Zie ook

[text.antiAliasSharpness](#), [text.antiAliasThickness](#)



## text.getTextAttr()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.getTextAttr(attrName [, startIndex [, endIndex]])
```

### Parameters

**attrName** Een tekenreeks met de naam van de objecteigenschap TextAttrs die moet worden geretourneerd. Voor een lijst met mogelijke waarden voor *attrName*, zie het overzicht met eigenschappen voor het [Object TextAttrs](#).

**startIndex** Een geheel getal dat de index van het eerste teken is. Deze parameter is optioneel.

**endIndex** Een geheel getal dat het einde aangeeft van het tekstbereik dat begint op *startIndex* en doorloopt tot aan, maar niet inclusief, *endIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

De waarde van het kenmerk dat is opgegeven in de parameter *attrName*.

### Beschrijving

Methode; het kenmerk dat met de parameter *attrName* is opgegeven, wordt opgehaald voor de tekst die wordt bepaald door de optionele parameters *startIndex* en *endIndex*. Als het kenmerk niet consistent is met het opgegeven bereik, wordt *undefined* geretourneerd. Als u de optionele parameters *startIndex* en *endIndex* weglaat, wordt het hele tekstbereik geretourneerd. Als u alleen *startIndex* opgeeft, bestaat het bereik alleen uit het teken op die positie. Als u zowel *startIndex* als *endIndex* opgeeft, begint het bereik op *startIndex* en loopt het door tot aan, maar niet inclusief, *endIndex*.

### Voorbeeld

In het volgende voorbeeld wordt de tekengrootte van het geselecteerde tekstveld opgehaald en weergegeven:

```
var TheTextSize = fl.getDocumentDOM().selection[0].getTextAttr("size");  
fl.trace(TheTextSize);
```

In het volgende voorbeeld wordt de tekstvulkleur van het geselecteerde tekstveld opgehaald:

```
var TheFill = fl.getDocumentDOM().selection[0].getTextAttr("fillColor");  
fl.trace(TheFill);
```

In het volgende voorbeeld wordt de grootte van het derde teken opgehaald:

```
var Char3 = fl.getDocumentDOM().selection[0].getTextAttr("size", 2);  
fl.trace(Char3);
```

In het volgende voorbeeld wordt de kleur opgehaald van het derde teken tot aan het achtste teken in het geselecteerde tekstveld:

```
fl.getDocumentDOM().selection[0].getTextAttr("fillColor", 2, 8);
```

## text.getTextString()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.getTextString([startIndex [, endIndex]])
```

### Parameters

**startIndex** Een geheel getal dat de index (op nul gebaseerd) van het eerste teken aangeeft. Deze parameter is optioneel.

**endIndex** Een geheel getal dat het einde aangeeft van het tekstbereik dat begint op *startIndex* en doorloopt tot aan, maar niet inclusief, *endIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Een tekenreeks met de tekst in het opgegeven bereik.

### Beschrijving

Methode; hiermee wordt het opgegeven tekstbereik opgehaald. Als u de optionele parameters *startIndex* en *endIndex* weglaat, wordt de hele tekenreeks geretourneerd. Als u alleen *startIndex* opgeeft, wordt de tekenreeks opgehaald die begint op de indexlocatie en eindigt bij het einde van het veld. Als u zowel *startIndex* als *endIndex* opgeeft, wordt de tekenreeks geretourneerd die begint op *startIndex* en doorloopt tot aan, maar niet inclusief, *endIndex*.

### Voorbeeld

In het volgende voorbeeld worden een of meer tekens opgehaald van het vijfde teken tot het einde van het geselecteerde tekstveld:

```
var myText = fl.getDocumentDOM().selection[0].getTextString(4);  
fl.trace(myText);
```

In het volgende voorbeeld worden het vierde tot het negende teken in het geselecteerde tekstveld opgehaald:

```
var myText = fl.getDocumentDOM().selection[0].getTextString(3, 9);  
fl.trace(myText);
```

## text.length

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.length
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat het aantal tekens in het object Text aangeeft.

### Voorbeeld

In het volgende voorbeeld wordt het aantal tekens in de geselecteerde tekst geretourneerd:

```
var textLength = fl.getDocumentDOM().selection[0].length;
```

## text.lineType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.lineType
```

### Beschrijving

Eigenschap; een tekenreeks waarmee het regeltype wordt ingesteld. Acceptabele waarden zijn "single line", "multiline", "multiline no wrap" en "password".

Deze eigenschap werkt alleen met dynamische tekst of invoertekst en genereert een waarschuwing bij statische tekst. De waarde "password" werkt alleen voor invoertekst.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `lineType` ingesteld op de waarde `multiline no wrap`:

```
fl.getDocumentDOM().selection[0].lineType = "multiline no wrap";
```

## text.maxCharacters

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.maxCharacters
```

### Beschrijving

Eigenschap; een geheel getal dat het maximale aantal tekens aangeeft dat de gebruiker kan invoeren in dit object Text.

Deze eigenschap werkt alleen met invoertekst; bij andere teksttypen wordt een waarschuwing gegenereerd.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `maxCharacters` ingesteld op 30:

```
fl.getDocumentDOM().selection[0].maxCharacters = 30;
```

## text.orientation

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.orientation
```

### Beschrijving

Eigenschap; een tekenreeks die de richting van het tekstveld aangeeft. Acceptabele waarden zijn "horizontal", "vertical left to right" en "vertical right to left".

Deze eigenschap werkt alleen met statische tekst; bij andere teksttypen wordt een waarschuwing gegenereerd.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap voor de tekstrichting ingesteld op `vertical right to left`:

```
fl.getDocumentDOM().selection[0].orientation = "vertical right to left";
```

## text.renderAsHTML

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.renderAsHTML
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Als de waarde `true` is, wordt de tekst weergegeven als HTML en worden ingesloten HTML-codes geïnterpreteerd.

Deze eigenschap werkt alleen met dynamische tekst of invoertekst; bij andere teksttypen wordt een waarschuwing gegenereerd.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `renderAsHTML` ingesteld op `true`:

```
fl.getDocumentDOM().selection[0].renderAsHTML = true;
```

## text.scrollable

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.scrollable
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Als de waarde `true` is, kan door de tekst worden geschoven.

Deze eigenschap werkt alleen met dynamische tekst of invoertekst; bij statische tekst wordt een waarschuwing gegenereerd.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `scrollable` ingesteld op `false`:

```
fl.getDocumentDOM().selection[0].scrollable = false;
```

## text.selectable

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.selectable
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Als de waarde `true` is, kan de tekst worden geselecteerd.

Invoertekst kan altijd worden geselecteerd. Er wordt een waarschuwing gegenereerd als deze eigenschap op `false` is ingesteld bij invoertekst.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `selectable` ingesteld op `true`:

```
fl.getDocumentDOM().selection[0].selectable = true;
```

## text.selectionEnd

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.selectionEnd
```

### Beschrijving

Eigenschap; een op nul gebaseerd geheel getal dat het einde van een subselectie in een tekst aangeeft. Zie [text.selectionStart](#) voor meer informatie.

## text.selectionStart

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.selectionStart
```

### Beschrijving

Eigenschap; een op nul gebaseerd geheel getal dat het begin van een subselectie in een tekst aangeeft. U kunt deze eigenschap met `text.selectionEnd` gebruiken om een bereik met tekens te selecteren. De tekens tot aan, maar niet inclusief, `text.selectionEnd` worden geselecteerd. Zie [text.selectionEnd](#).

- Als er een invoegpunt of geen selectie aanwezig is, is `text.selectionEnd` gelijk aan `text.selectionStart`.
- Als `text.selectionStart` op een hogere waarde dan `text.selectionEnd` is ingesteld, wordt `text.selectionEnd` ingesteld op `text.selectionStart` en wordt er geen tekst geselecteerd.

### Voorbeeld

In het volgende voorbeeld wordt het begin van de subselectie in de tekst ingesteld op het zesde teken:

```
fl.getDocumentDOM().selection[0].selectionStart = 5;
```

In het volgende voorbeeld worden de tekens Barbara geselecteerd in een tekstveld dat de tekst `My name is Barbara` bevat en worden deze vet en groen opgemaakt:

```
fl.getDocumentDOM().selection[0].selectionStart = 11;  
fl.getDocumentDOM().selection[0].selectionEnd = 18;  
var s = fl.getDocumentDOM().selection[0].selectionStart;  
var e = fl.getDocumentDOM().selection[0].selectionEnd;  
fl.getDocumentDOM().setElementTextAttr('bold', true, s, e);  
fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00", s, e);
```

## text.setTextAttr()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.setTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

### Parameters

**attrName** Een tekenreeks met de naam van de objecteigenschap `TextAttrs` die moet worden gewijzigd.

**attrValue** De waarde voor de objecteigenschap `TextAttrs`.

Voor een lijst met mogelijke waarden voor `attrName` en `attrValue`, zie het overzicht met eigenschappen voor het [Object TextAttrs](#).

**startIndex** Een geheel getal dat de index (op nul gebaseerd) van het eerste teken in de array aangeeft. Deze parameter is optioneel.

**endIndex** Een geheel getal dat de index bevat van het eindpunt in de geselecteerde tekenreeks die begint op *startIndex* en doorloopt tot aan, maar niet inclusief, *endIndex*. Deze parameter is optioneel.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; het kenmerk dat wordt opgegeven met de parameter *attrName* voor de tekst die door *startIndex* en *endIndex* wordt bepaald, wordt ingesteld op de waarde die wordt opgegeven met *attrValue*. U kunt deze methode gebruiken om kenmerken te wijzigen van tekst die TextRun-elementen bevat (zie [Object TextRun](#)), of die deel uitmaakt van bestaande TextRun-elementen. Hierdoor kunnen de positie van de TextRun-elementen en het aantal TextRun-elementen in de array van dit object worden gewijzigd (zie [text.textRun.textRuns](#)).

Als u de optionele parameters weglaat, wordt het hele tekenbereik van het object Text gebruikt. Als u alleen *startIndex* opgeeft, bestaat het bereik alleen uit het teken op die positie. Als u zowel *startIndex* als *endIndex* opgeeft, begint het bereik op *startIndex* en loopt het door tot aan, maar niet inclusief, het teken op *endIndex*.

#### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde tekstveld cursief gemaakt:

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

In het volgende voorbeeld wordt de grootte van het derde teken op 10 ingesteld:

```
fl.getDocumentDOM().selection[0].setTextAttr("size", 10, 2);
```

In het volgende voorbeeld wordt de kleur voor het derde tot aan het achtste teken van de geselecteerde tekst op rood ingesteld:

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

## text.setTextString()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
text.setTextString(text [, startIndex [, endIndex]])
```

#### Parameters

**text** Een tekenreeks die de tekens bevat die in dit object Text moeten worden ingevoegd.

**startIndex** Een geheel getal dat de index (op nul gebaseerd) aangeeft van het teken in de tekenreeks waar de tekst wordt ingevoegd. Deze parameter is optioneel.

**endIndex** Een geheel getal dat de index aangeeft van het eindpunt in de geselecteerde tekst. De nieuwe tekst overschrijft de tekst van *startIndex* tot aan, maar niet inclusief, *endIndex*. Deze parameter is optioneel.

#### Geretourneerde waarde

Niets.

### Beschrijving

Eigenschap; hiermee wordt de teksttekenreeks in dit object Text gewijzigd. Als u de optionele parameters weglaat, wordt het hele object Text vervangen. Als u alleen *startIndex* opgeeft, wordt de opgegeven tekenreeks ingevoegd op de positie van *startIndex*. Als u zowel *startIndex* als *endIndex* opgeeft, vervangt de doorgegeven tekenreeks het tekstsegment van *startIndex* tot aan, maar niet inclusief, *endIndex*.

### Voorbeeld

In het volgende voorbeeld wordt de tekenreeks `this is a string` toegewezen aan het geselecteerde tekstveld:

```
fl.getDocumentDOM().selection[0].setTextString("this is a string");
```

In het volgende voorbeeld wordt de tekenreeks `abc` ingevoegd bij het vijfde teken van het geselecteerde tekstveld:

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");  
fl.getDocumentDOM().selection[0].setTextString("abc", 4);  
// text field is now "0123abc4567890"
```

In het volgende voorbeeld wordt de tekst van het derde tot aan het achtste teken van de geselecteerde tekst vervangen door de tekenreeks `abcdefghij`. De tekens tussen *startIndex* en *endIndex* worden overschreven. De tekens die op *endIndex* beginnen, volgen na de ingevoegde tekenreeks.

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");  
fl.getDocumentDOM().selection[0].setTextString("abcdefghij", 2, 8);  
// text field is now "01abcdefghij890"
```

## text.shortcut

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.shortcut
```

### Beschrijving

Eigenschap; een tekenreeks die gelijk is aan het veld Sneltoets in het deelvenster Toegankelijkheid. De sneltoets wordt gelezen door de schermlezer. Deze eigenschap kan niet worden gebruikt met dynamische tekst.

### Voorbeeld

In het volgende voorbeeld wordt de sneltoets van het geselecteerde object opgehaald en wordt de waarde weergegeven:

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;  
fl.trace(theShortcut);
```

In het volgende voorbeeld wordt de sneltoets van het geselecteerde object ingesteld:

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```



## text.silent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.silent
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft of het object toegankelijk is. Dit is gelijk aan de inverse logica van de instelling Object toegankelijk maken in het deelvenster Toegankelijkheid. Dit betekent dat als `silent` op `true` is ingesteld, Object toegankelijk maken wordt uitgeschakeld. Als `false` is ingesteld, wordt Object toegankelijk maken ingeschakeld.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of het object toegankelijk is (de waarde `false` betekent dat het toegankelijk is):

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

In het volgende voorbeeld wordt het object toegankelijk gemaakt:

```
fl.getDocumentDOM().selection[0].silent = false;
```

## text.tabIndex

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
text.tabIndex
```

### Beschrijving

Eigenschap; een geheel getal dat gelijk is aan het veld Tabindex in het deelvenster Toegankelijkheid. Met deze waarde kunt u de volgorde bepalen waarin objecten worden benaderd wanneer de gebruiker op de Tab-toets drukt.

### Voorbeeld

In het volgende voorbeeld wordt de `tabIndex` van het geselecteerde object opgehaald:

```
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;
```

In het volgende voorbeeld wordt de `tabIndex` van het geselecteerde object ingesteld:

```
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

## text.textRuns

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.textRuns`

### Beschrijving

Alleen-lezen eigenschap; een array met TextRun-objecten (zie [Object TextRun](#)).

### Voorbeeld

In het volgende voorbeeld wordt de waarde van de eigenschap `textRuns` opgeslagen in de variabele `myTextRuns`:

```
var myTextRuns = fl.getDocumentDOM().selection[0].textRuns;
```

## text.textType

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.textType`

### Beschrijving

Eigenschap; een tekenreeks die het type van het tekstveld aangeeft. Acceptabele waarden zijn "static", "dynamic" en "input".

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `textType` ingesteld op `input`:

```
fl.getDocumentDOM().selection[0].textType = "input";
```

## text.useDeviceFonts

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.useDeviceFonts`

### Beschrijving

Eigenschap; een Booleaanse waarde. Als de waarde `true` is, wordt tekst weergegeven met apparaatlettertypen.

### Voorbeeld

In het volgende voorbeeld worden apparaatlettertypen gebruikt voor het weergeven van tekst:

```
fl.getDocumentDOM().selection[0].useDeviceFonts = true;
```

## text.variableName

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`text.variableName`

### Beschrijving

Eigenschap; een tekenreeks die de naam bevat van de variabele die is gekoppeld aan het object Text. Deze eigenschap werkt alleen met dynamische tekst of invoertekst; bij andere teksttypen wordt een waarschuwing gegenereerd.

Deze eigenschap wordt alleen ondersteund in ActionScript 1.0 en ActionScript 2.0.

### Voorbeeld

In het volgende voorbeeld wordt de variabelenaam van het geselecteerde tekstvak ingesteld op `firstName`:

```
fl.getDocumentDOM().selection[0].variableName = "firstName";
```

# Hoofdstuk 43: Object TextAttrs

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object TextAttrs bevat alle teksteigenschappen die op een subselectie kunnen worden toegepast. Dit object is een eigenschap van het object TextRun (`textRun.textAttrs`).

## Overzicht van eigenschappen

Voor het object TextAttrs zijn de volgende eigenschappen beschikbaar:

Eigenschap	Beschrijving
<code>textAttrs.aliasText</code>	Een Booleaanse waarde die aangeeft dat de tekst moet worden getekend met een methode die is geoptimaliseerd om de leesbaarheid van kleine tekst te vergroten.
<code>textAttrs.alignment</code>	Een tekenreeks die de uitvulling van alinea's aangeeft. Acceptabele waarden zijn "left", "center", "right" en "justify".
<code>textAttrs.autoKern</code>	Een Booleaanse waarde die bepaalt of voor het spatiëren de informatie over paartekenspatiëring in het lettertype wordt gebruikt ( <code>true</code> ) of genegeerd ( <code>false</code> ).
<code>textAttrs.bold</code>	Een Booleaanse waarde. Bij de waarde <code>true</code> verschijnt het lettertype vetgedrukt.
<code>textAttrs.characterPosition</code>	Een tekenreeks die de basislijn voor de tekst bepaalt.
<code>textAttrs.characterSpacing</code>	Afgekeurd en vervangen door <code>textAttrs.letterSpacing</code> . Een geheel getal dat de spatiëring tussen tekens vertegenwoordigt.
<code>textAttrs.face</code>	Een tekenreeks die de naam van het lettertype vertegenwoordigt, bijvoorbeeld "Arial".
<code>textAttrs.fillColor</code>	Een tekenreeks, hexadecimale waarde of geheel getal dat de vulkleur vertegenwoordigt.
<code>textAttrs.indent</code>	Een geheel getal dat de inspringing van de alinea aangeeft.
<code>textAttrs.italic</code>	Een Booleaanse waarde. Bij de waarde <code>true</code> verschijnt het lettertype cursief gedrukt.
<code>textAttrs.leftMargin</code>	Een geheel getal dat de linkermarge van de alinea aangeeft.
<code>textAttrs.letterSpacing</code>	Een geheel getal dat de spatiëring tussen tekens vertegenwoordigt.
<code>textAttrs.lineSpacing</code>	Een geheel getal dat de regelafstand (voorloop) van de alinea aangeeft.
<code>textAttrs.rightMargin</code>	Een geheel getal dat de rechtermarge van de alinea aangeeft.
<code>textAttrs.rotation</code>	Een Booleaanse waarde. Bij de waarde <code>true</code> worden de tekens van de tekst met 90° geroteerd. De standaardwaarde is <code>false</code> .
<code>textAttrs.size</code>	Een geheel getal dat de grootte van het lettertype aangeeft.
<code>textAttrs.target</code>	Een tekenreeks die de eigenschap <code>target</code> van het tekstveld vertegenwoordigt.
<code>textAttrs.url</code>	Een tekenreeks die de eigenschap <code>URL</code> van het tekstveld vertegenwoordigt.

## textAttrs.aliasText

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.aliasText
```

### Beschrijving

Eigenschap; een Booleaanse waarde die aangeeft dat de tekst moet worden getekend met een methode die is geoptimaliseerd om de leesbaarheid van kleine tekst te vergroten.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `aliasText` voor alle tekst in het momenteel geselecteerde tekstveld op `true` gezet:

```
fl.getDocumentDOM().setElementTextAttr('aliasText', true);
```

## textAttrs.alignment

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.alignment
```

### Beschrijving

Eigenschap; een tekenreeks die de uitvulling van alinea's bepaalt. Acceptabele waarden zijn "left", "center", "right" en "justify".

### Voorbeeld

In het volgende voorbeeld wordt ingesteld dat alinea's die tekens bevatten van index 0 tot, maar niet inclusief, index 3, moeten worden uitgevuld. Dit kan ook van invloed zijn op tekens buiten het opgegeven bereik die in dezelfde alinea staan.

```
fl.getDocumentDOM().setTextSelection(0, 3);  
fl.getDocumentDOM().setElementTextAttr("alignment", "justify");
```

## textAttrs.autoKern

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.autoKern
```

### Beschrijving

Eigenschap; een Booleaanse waarde die bepaalt of voor het spatiëren de informatie over paartekenspatiëring in het lettertype wordt gebruikt (`true`) of genegeerd (`false`).

### Voorbeeld

In het volgende voorbeeld worden de tekens van index 2 tot, maar niet inclusief, index 6 geselecteerd en wordt de eigenschap `autoKern` ingesteld op `true`:

```
fl.getDocumentDOM().setTextSelection(3, 6);  
fl.getDocumentDOM().setElementTextAttr('autoKern', true);
```

## textAttrs.bold

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.bold
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Bij de waarde `true` verschijnt het lettertype vetgedrukt.

### Voorbeeld

In het volgende voorbeeld wordt het eerste teken van het geselecteerde object `Text` geselecteerd en wordt de eigenschap `bold` ingesteld op `true`:

```
fl.getDocumentDOM().setTextSelection(0, 1);  
fl.getDocumentDOM().setElementTextAttr('bold', true);
```

## textAttrs.characterPosition

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.characterPosition
```

### Beschrijving

Eigenschap; een tekenreeks die de basislijn voor de tekst bepaalt. Acceptabele waarden zijn `"normal"`, `"subscript"` en `"superscript"`. Deze eigenschap wordt alleen toegepast op statische tekst.

### Voorbeeld

In het volgende voorbeeld worden de tekens van het geselecteerde tekstveld van index 2 tot, maar niet inclusief, index 6 geselecteerd en wordt de eigenschap `characterPosition` ingesteld op `subscript`:

```
fl.getDocumentDOM().setTextSelection(2, 6);  
fl.getDocumentDOM().setElementTextAttr("characterPosition", "subscript");
```

## textAttrs.characterSpacing

### Beschikbaarheid

Flash MX 2004. In Flash 8 afgekeurd en vervangen door `textAttrs.letterSpacing`.

### Gebruik

```
textAttrs.characterSpacing
```

### Beschrijving

Eigenschap; een geheel getal dat de spatiëring tussen tekens vertegenwoordigt. Acceptabele waarden zijn -60 tot en met 60.

Deze eigenschap werkt alleen met statische tekst; bij gebruik met andere teksttypes wordt een waarschuwingsbericht gegenereerd.

### Voorbeeld

In het volgende voorbeeld wordt de spatiëring van het geselecteerde tekstveld ingesteld op 10:

```
fl.getDocumentDOM().setElementTextAttr("characterSpacing", 10);
```

## textAttrs.face

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.face
```

### Beschrijving

Eigenschap; een tekenreeks die de naam van het lettertype vertegenwoordigt, bijvoorbeeld "Arial".

### Voorbeeld

In het volgende voorbeeld wordt het lettertype in het geselecteerde tekstveld van het teken bij index 2 tot, maar niet inclusief, het teken bij index 8 ingesteld op Arial:

```
fl.getDocumentDOM().selection[0].setTextAttr("face", "Arial", 2, 8);
```

## textAttrs.fillColor

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.fillColor
```

### Beschrijving

Eigenschap; de kleur van de vulling in een van de volgende notaties:

- Een tekenreeks in de notatie "#RRGGBB" of "#RRGGBBAA"
- Een hexadecimaal getal in de notatie 0xRRGGBB
- Een geheel getal dat het decimale equivalent is van een hexadecimaal getal

### Voorbeeld

In het volgende voorbeeld wordt de kleur rood ingesteld voor het geselecteerde tekstveld van het teken bij index 2 tot, maar niet inclusief, het teken bij index 8:

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

## textAttrs.indent

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.indent
```

### Beschrijving

Eigenschap; een geheel getal dat de inspringing van de alinea aangeeft. Acceptabele waarden zijn -720 tot en met 720.

### Voorbeeld

In het volgende voorbeeld wordt de inspringing van het geselecteerde tekstveld van het teken bij index 2 tot, maar niet inclusief, het teken bij index 8 ingesteld op 100. Dit kan ook van invloed zijn op tekens buiten het opgegeven bereik die in dezelfde alinea staan.

```
fl.getDocumentDOM().selection[0].setTextAttr("indent", 100, 2, 8);
```

## textAttrs.italic

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.italic
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Bij de waarde `true` verschijnt het lettertype cursief gedrukt.

### Voorbeeld

In het volgende voorbeeld wordt het geselecteerde tekstveld cursief gemaakt:

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```



## textAttrs.leftMargin

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.leftMargin
```

### Beschrijving

Eigenschap; een geheel getal dat de linkermarge van de paragraaf aangeeft. Acceptabele waarden zijn 0 tot en met 720.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `leftMargin` van het geselecteerde tekstveld van het teken bij index 2 tot, maar niet inclusief, het teken bij index 8 ingesteld op 100. Dit kan ook van invloed zijn op tekens buiten het opgegeven bereik die in dezelfde alinea staan.

```
fl.getDocumentDOM().selection[0].setTextAttr("leftMargin", 100, 2, 8);
```

## textAttrs.letterSpacing

### Beschikbaarheid

Flash 8.

### Gebruik

```
textAttrs.letterSpacing
```

### Beschrijving

Eigenschap; een geheel getal dat de spatiëring tussen tekens vertegenwoordigt. Acceptabele waarden zijn -60 tot en met 60.

Deze eigenschap werkt alleen met statische tekst; bij gebruik met andere teksttypes wordt een waarschuwingsbericht gegenereerd.

### Voorbeeld

In de volgende code worden de tekens van index 0 tot, maar niet inclusief, index 10 geselecteerd en wordt de spatiëring ingesteld op 60:

```
fl.getDocumentDOM().setTextSelection(0, 10);  
fl.getDocumentDOM().setElementTextAttr("letterSpacing", 60);
```

## textAttrs.lineSpacing

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.lineSpacing
```

### Beschrijving

Eigenschap; een geheel getal dat de regelafstand (*voorloop*) van de alinea aangeeft. Acceptabele waarden zijn -360 tot en met 720.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `lineSpacing` van het geselecteerde tekstveld ingesteld op 100:

```
fl.getDocumentDOM().selection[0].setTextAttr("lineSpacing", 100);
```

## textAttrs.rightMargin

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.rightMargin
```

### Beschrijving

Eigenschap; een geheel getal dat de rechtermarge van de alinea aangeeft. Acceptabele waarden zijn 0 tot en met 720.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `rightMargin` van het geselecteerde tekstveld van het teken bij index 2 tot, maar niet inclusief, het teken bij index 8 ingesteld op 100. Dit kan ook van invloed zijn op tekens buiten het opgegeven bereik die in dezelfde alinea staan.

```
fl.getDocumentDOM().selection[0].setTextAttr("rightMargin", 100, 2, 8);
```

## textAttrs.rotation

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.rotation
```

### Beschrijving

Eigenschap; een Booleaanse waarde. Bij de waarde `true` worden de tekens van de tekst met 90° geroteerd. De standaardwaarde is `false`. Deze eigenschap werkt alleen met statische tekst met een verticale richting; bij gebruik met andere teksttypes wordt een waarschuwingsbericht gegenereerd.

### Voorbeeld

In het volgende voorbeeld wordt de rotatie van het geselecteerde tekstveld ingesteld op `true`:

```
fl.getDocumentDOM().setElementTextAttr("rotation", true);
```

## textAttrs.size

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.size
```

### Beschrijving

Eigenschap; een geheel getal dat de grootte van het lettertype aangeeft.

### Voorbeeld

In het volgende voorbeeld wordt de grootte van het teken met index 2 opgevraagd en verschijnt het resultaat in het deelvenster Uitvoer:

```
fl.outputPanel.trace(fl.getDocumentDOM().selection[0].getTextAttr("size", 2));
```

## textAttrs.target

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.target
```

### Beschrijving

Eigenschap; een tekenreeks die de eigenschap `target` van het tekstveld vertegenwoordigt. Deze eigenschap werkt alleen met statische tekst.

### Voorbeeld

In het volgende voorbeeld wordt de eigenschap `target` van het tekstveld in het eerste frame van de toplaag van de huidige scène opgehaald en weergegeven in het deelvenster Uitvoer:

```
fl.outputPanel.trace(fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].getTextAttr("target"));
```

## textAttrs.url

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textAttrs.url
```

### **Beschrijving**

Eigenschap; een tekenreeks die de eigenschap URL van het tekstveld vertegenwoordigt. Deze eigenschap werkt alleen met statische tekst.

### **Voorbeeld**

In het volgende voorbeeld wordt de URL van het geselecteerde tekstveld ingesteld op `http://www.adobe.com`:

```
fl.getDocumentDOM().setElementTextAttr("url", "http://www.adobe.com");
```

# Hoofdstuk 44: Object TextRun

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object TextRun vertegenwoordigt een groep opeenvolgende tekens met attributen die overeenkomen met alle eigenschappen in het [Object TextAttrs](#). Dit object is een eigenschap van het object Text (`text.textRuns`).

## Overzicht van eigenschappen

Naast de eigenschappen die beschikbaar zijn voor gebruik met het object Text, kunt u de volgende eigenschappen gebruiken voor het object TextRun:

Eigenschap	Beschrijving
<code>textRun.characters</code>	Een tekenreeks die de tekst vertegenwoordigt die het object TextRun bevat.
<code>textRun.textAttrs</code>	Het object TextAttrs dat de attributen van de tekstregels bevat.

## textRun.textAttrs

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
textRun.textAttrs
```

### Beschrijving

Eigenschap; het [Object TextAttrs](#) dat de attributen van de tekstregels bevat.

### Voorbeeld

In het volgende voorbeeld worden de eigenschappen van de eerste groep tekens in het geselecteerde tekstveld weergegeven in het deelvenster Uitvoer.

```
var curTextAttrs = fl.getDocumentDOM().selection[0].textRuns[0].textAttrs;
for (var prop in curTextAttrs) {
    fl.trace(prop + " = " + curTextAttrs[prop]);
}
```

## textRun.characters

### Beschikbaarheid

Flash MX 2004.

### **Gebruik**

`textRun.characters`

### **Beschrijving**

Eigenschap; de tekst die het object TextRun bevat.

### **Voorbeeld**

In het volgende voorbeeld worden de tekens die de eerste groep tekens in het geselecteerde tekstveld bevat, weergegeven in het deelvenster Uitvoer.

```
fl.trace(fl.getDocumentDOM().selection[0].textRuns[0].characters);
```

# Hoofdstuk 45: Object Timeline

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Timeline staat voor de Flash-tijdlijn, die voor het huidige document toegankelijk is met `fl.getDocumentDOM().getTimeline()`. Deze methode retourneert de tijdlijn van de huidige scène of het symbool dat wordt bewerkt.

Wanneer u met scènes werkt, heeft de tijdlijn van elke scène een indexwaarde en is deze voor het huidige document toegankelijk met `fl.getDocumentDOM().timelines[i]`. (In dit voorbeeld is `i` de index van de waarde van de tijdlijn.)

Wanneer u met frames werkt met behulp van de methoden en eigenschappen van het object Timeline, moet u er rekening mee houden dat de framewaarde een op nul gebaseerde index is (niet het daadwerkelijke framenummer in de reeks frames op de tijdlijn). Dat wil zeggen dat het eerste frame de frame-index 0 heeft.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Timeline:

Methode	Beschrijving
<code>timeline.addMotionGuide()</code>	Hiermee wordt een geleidelaag voor beweging toegevoegd boven de huidige laag en wordt de huidige laag aan de nieuwe geleidelaag gekoppeld.
<code>timeline.addNewLayer()</code>	Hiermee wordt een nieuwe laag aan het document toegevoegd en ingesteld als huidige laag.
<code>timeline.clearFrames()</code>	Hiermee wordt alle inhoud verwijderd uit een frame of een reeks frames op de huidige laag.
<code>timeline.clearKeyframes()</code>	Hiermee wordt een hoofdfraam geconverteerd naar een regulier frame en wordt de inhoud ervan op de huidige laag verwijderd.
<code>timeline.convertToBlankKeyframes()</code>	Hiermee worden frames geconverteerd naar lege hoofdfraam op de huidige laag.
<code>timeline.convertToKeyframes()</code>	Hiermee wordt een reeks geconverteerd naar hoofdfraam (of wordt de selectie geconverteerd als er geen frames zijn opgegeven) op de huidige laag.
<code>timeline.copyFrames()</code>	Hiermee wordt een reeks frames op de huidige laag naar het klembord gekopieerd.
<code>timeline.copyLayers()</code>	Hiermee wordt een reeks tijdlijnlagen gekopieerd naar het klembord.
<code>timeline.copyMotion()</code>	Hiermee wordt beweging op geselecteerde frames gekopieerd, hetzij van een bewegingstween of van een frame-voor-frame animatie, zodat deze op andere frames kan worden toegepast.
<code>timeline.copyMotionAsAS3()</code>	Hiermee wordt beweging op geselecteerde frames, hetzij van een bewegings-tween of van een frame-voor-frame animatie, als ActionScript 3.0-code gekopieerd naar het klembord.
<code>timeline.createMotionObject()</code>	Maakt een nieuw bewegingsobject bij een toegewezen begin- en eindframe.

Methode	Beschrijving
<code>timeline.createMotionTween()</code>	Hiermee wordt de eigenschap <code>frame.tweenType</code> voor elk geselecteerd hoofdfame op de huidige laag ingesteld op <code>motion</code> en wordt de inhoud van alle frames zo nodig geconverteerd naar één symbolinstantie.
<code>timeline.cutFrames()</code>	Hiermee wordt een reeks frames op de huidige laag van de tijdlijn geknipt en op het klembord opgeslagen.
<code>timeline.cutLayers()</code>	Hiermee wordt een reeks tijdlijnlagen geknipt en opgeslagen naar het klembord.
<code>timeline.deleteLayer()</code>	Hiermee wordt een laag verwijderd.
<code>timeline.duplicateLayers()</code>	Hiermee worden de geselecteerde lagen of opgegeven lagen gedupliceerd.
<code>timeline.expandFolder()</code>	Hiermee worden de opgegeven map of mappen uit- of samengevouwen.
<code>timeline.findLayerIndex()</code>	Hiermee wordt gezocht naar een array met indexen voor de lagen met de opgegeven naam.
<code>timeline.getFrameProperty()</code>	Hiermee wordt de waarde van de opgegeven eigenschap voor de geselecteerde frames opgehaald.
<code>timeline.getGuidelines()</code>	Hiermee wordt een XML-tekenreeks geretourneerd die de huidige posities van de horizontale en verticale hulplijnen voor een tijdlijn vertegenwoordigt (Weergave > Hulplijnen >Hulplijnen weergeven).
<code>timeline.getLayerProperty()</code>	Hiermee wordt de waarde van de opgegeven eigenschap voor de geselecteerde lagen opgehaald.
<code>timeline.getSelectedFrames()</code>	Hiermee worden de momenteel geselecteerde frames in een array opgehaald.
<code>timeline.getSelectedLayers()</code>	Hiermee worden de op nul gebaseerde indexwaarden van de momenteel geselecteerde lagen opgehaald.
<code>timeline.insertBlankKeyframe()</code>	Hiermee wordt in de opgegeven frame-index een leeg hoofdfame ingevoegd. Als de index niet wordt opgegeven, wordt het lege hoofdfame ingevoegd aan de hand van de afspeelkop of de selectie.
<code>timeline.insertFrames()</code>	Hiermee wordt het opgegeven aantal frames ingevoegd bij het opgegeven framenummer.
<code>timeline.insertKeyframe()</code>	Hiermee wordt een hoofdfame ingevoegd bij het opgegeven frame.
<code>timeline.pasteFrames()</code>	Hiermee wordt de reeks frames van het klembord in de opgegeven frames geplakt.
<code>timeline.pasteLayers()</code>	Hiermee worden de gekopieerde lagen geplakt naar de tijdlijn die boven de opgegeven laagindex ligt.
<code>timeline.pasteMotion()</code>	Hiermee wordt het bereik met bewegingsframes dat door <code>timeline.copyMotion()</code> is opgehaald op de tijdlijn geplakt.
<code>timeline.removeFrames()</code>	Hiermee wordt het frame verwijderd.
<code>timeline.removeMotionObject()</code>	Verwijdert het bewegingsobject dat is gemaakt met <code>timeline.createMotionObject()</code> en zet het frame of de frames om in statische frames.
<code>timeline.reorderLayer()</code>	Hiermee wordt de eerste opgegeven laag voor of achter de tweede opgegeven laag geplaatst.
<code>timeline.reverseFrames()</code>	Hiermee wordt een reeks frames omgekeerd.



Methode	Beschrijving
<code>timeline.selectAllFrames()</code>	Hiermee worden alle frames in de huidige tijdlijn geselecteerd.
<code>timeline setFrameProperty()</code>	Hiermee wordt de eigenschap van het object Frame ingesteld voor de geselecteerde frames.
<code>timeline.setGuidelines()</code>	Hiermee worden de hulplijnen voor de tijdlijn vervangen door de opgegeven informatie.
<code>timeline.setLayerProperty()</code>	Hiermee wordt de opgegeven eigenschap voor alle geselecteerde lagen ingesteld op een opgegeven waarde.
<code>timeline.setSelectedFrames()</code>	Hiermee wordt een reeks frames in de huidige laag geselecteerd of worden de geselecteerde frames ingesteld in de selectiearray die wordt doorgegeven aan deze methode.
<code>timeline.setSelectedLayers()</code>	Hiermee wordt ingesteld welke laag moet worden geselecteerd en wordt de opgegeven laag de huidige laag.
<code>timeline.showLayerMasking()</code>	Hiermee wordt de maskering van de laag tijdens het ontwerpen weergegeven door het masker en de gemaskeerde lagen te vergrendelen.
<code>timeline.startPlayback()</code>	Hiermee wordt het automatisch afspelen van de tijdlijn gestart als deze op het moment niet wordt afgespeeld.
<code>timeline.stopPlayback()</code>	Hiermee wordt het automatisch afspelen van de tijdlijn gestopt als deze op het moment wordt afgespeeld.

### Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Timeline:

Eigenschap	Beschrijving
<code>timeline.currentFrame</code>	Een op nul gebaseerde index voor het frame op de huidige locatie van de afspreekop.
<code>timeline.currentLayer</code>	Een op nul gebaseerde index voor de momenteel actieve laag.
<code>timeline.frameCount</code>	Alleen-lezen; een geheel getal dat staat voor het aantal frames op de langste laag van deze tijdlijn.
<code>timeline.layerCount</code>	Alleen-lezen; een geheel getal dat staat voor het aantal lagen op de opgegeven tijdlijn.
<code>timeline.layers</code>	Alleen-lezen; een array met laagobjecten.
<code>timeline.libraryItem</code>	Alleen-lezen eigenschap; geeft aan of de tijdlijn bij een scène hoort.
<code>timeline.name</code>	Een tekenreeks met de naam van de huidige tijdlijn.

## timeline.addMotionGuide()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.addMotionGuide()
```

### Parameters

Geen.

### Geretourneerde waarde

Een geheel getal dat de op nul gebaseerde index van de toegevoegde geleidelaag vertegenwoordigt. Wanneer de huidige laag niet van het type "Normal" is, dan retourneert Flash -1.

### Beschrijving

Methode; hiermee wordt een geleidelaag boven de huidige laag toegevoegd en wordt de huidige laag gekoppeld aan de toegevoegde geleidelaag, waardoor de huidige laag wordt geconverteerd naar het type "Guided" .

Deze methode werkt uitsluitend met een laag van het type "Normal". Op lagen van het type "Folder", "Mask", "Masked", "Guide" of "Guided" heeft ze geen effect.

### Voorbeeld

In het volgende voorbeeld wordt een geleidelaag toegevoegd boven de huidige laag en wordt de huidige laag geconverteerd naar Guided:

```
fl.getDocumentDOM().getTimeline().addMotionGuide();
```

## timeline.addNewLayer()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.addNewLayer([name] [, layerType [, bAddAbove]])
```

### Parameters

**name** Een tekenreeks die de naam van de nieuwe laag aangeeft. Wanneer u deze parameter niet instelt, krijgt de nieuwe laag een nieuwe standaardnaam ("Layer n", waarbij *n* het totale aantal lagen voorstelt). Deze parameter is optioneel.

**layerType** Een tekenreeks die het type aangeeft van de laag die u wilt toevoegen. Wanneer u deze parameter niet instelt, wordt een laag van het type "Normal" gemaakt. Deze parameter is optioneel. Acceptabele waarden zijn "normal", "guide", "guided", "mask", "masked" en "folder".

**bAddAbove** Een Booleaanse waarde: bij `true` (standaard) wordt de nieuwe laag boven de huidige laag toegevoegd; bij `false` wordt de nieuwe laag onder de huidige laag toegevoegd. Deze parameter is optioneel.

### Geretourneerde waarde

Een geheel getal van de op nul gebaseerde index van de toegevoegde laag.

### Beschrijving

Methode; hiermee wordt een nieuwe laag aan het document toegevoegd en ingesteld als huidige laag.

### Voorbeeld

In het volgende voorbeeld wordt een nieuwe laag met een door Flash gegenereerde standaardnaam toegevoegd aan de tijdlijn:

```
fl.getDocumentDOM().getTimeline().addNewLayer();
```

In het volgende voorbeeld wordt een nieuwe maplaag toegevoegd boven op de huidige laag. De maplaag krijgt de naam Folder1:

```
fl.getDocumentDOM().getTimeline().addNewLayer("Folder1", "folder", true);
```

## timeline.clearFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.clearFrames([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het begin vastlegt van de reeks frames die moet worden gewist. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die het einde vastlegt van de reeks frames die moet worden gewist. De reeks loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt alle inhoud verwijderd uit een frame of een reeks frames op de huidige laag.

### Voorbeeld

In het volgende voorbeeld worden de frames vanaf Frame 6 tot, maar niet inclusief, Frame 11 gewist (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().clearFrames(5, 10);
```

In het volgende voorbeeld wordt Frame 15 gewist:

```
fl.getDocumentDOM().getTimeline().clearFrames(14);
```

## timeline.clearKeyframes()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.clearKeyframes([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het begin vastlegt van de reeks frames die moet worden gewist. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die het einde vastlegt van de reeks frames die moet worden gewist. De reeks loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een hoofdframe geconverteerd naar een regulier frame en wordt de inhoud ervan op de huidige laag verwijderd.

### Voorbeeld

In het volgende voorbeeld worden de hoofdframes vanaf Frame 5 tot, maar niet inclusief, Frame 10 gewist (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().clearKeyframes(4, 9);
```

In het volgende voorbeeld wordt het hoofdframe op Frame 15 gewist en geconverteerd naar een regulier frame:

```
fl.getDocumentDOM().getTimeline().clearKeyframes(14);
```

## timeline.convertToBlankKeyframes()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.convertToBlankKeyframes([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het eerste frame aangeeft dat wordt geconverteerd naar een hoofdframe. Als u *startFrameIndex* niet instelt, converteert de methode de momenteel geselecteerde frames. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame de conversie naar hoofdframes wordt stopgezet. De reeks te converteren frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden frames geconverteerd naar lege hoofdframes op de huidige laag.

### Voorbeeld

In het volgende voorbeeld wordt Frame 2 tot, maar niet inclusief, Frame 10 geconverteerd naar een leeg hoofdfame (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(1, 9);
```

In het volgende voorbeeld wordt Frame 5 geconverteerd naar een leeg hoofdfame:

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(4);
```

## timeline.convertToKeyframes()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.convertToKeyframes([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het eerste frame aangeeft dat wordt geconverteerd naar een hoofdfame. Als u *startFrameIndex* niet instelt, converteert de methode de momenteel geselecteerde frames. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame de conversie naar hoofdfames wordt stopgezet. De reeks te converteren frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een reeks geconverteerd naar hoofdfames (of wordt de selectie geconverteerd als er geen frames zijn opgegeven) op de huidige laag.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde frames geconverteerd naar hoofdfames:

```
fl.getDocumentDOM().getTimeline().convertToKeyframes();
```

In het volgende voorbeeld worden de frames vanaf Frame 2 tot, maar niet inclusief, Frame 10 geconverteerd naar hoofdfames (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(1, 9);
```

In het volgende voorbeeld wordt Frame 5 geconverteerd naar een hoofdfame:

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(4);
```

## timeline.copyFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.copyFrames([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het begin aangeeft van de reeks frames die moet worden gekopieerd. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame het kopiëren wordt stopgezet. De reeks te kopiëren frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode: hiermee wordt een reeks frames op de huidige laag naar het klembord gekopieerd.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde frames gekopieerd naar het klembord:

```
fl.getDocumentDOM().getTimeline().copyFrames();
```

In het volgende voorbeeld wordt Frame 2 tot, maar niet inclusief, Frame 10 gekopieerd naar het klembord (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().copyFrames(1, 9);
```

In het volgende voorbeeld wordt Frame 5 gekopieerd naar het klembord:

```
fl.getDocumentDOM().getTimeline().copyFrames(4);
```

## timeline.copyLayers()

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
timeline.copyLayers([startLayerIndex [, endLayerIndex]])
```

### Parameters

**startLayerIndex** Optioneel. Een op nul gebaseerde index die het begin aangeeft van de reeks lagen die moet worden gekopieerd. Als u *startLayerIndex* niet instelt, gebruikt de methode de huidige selectie.

**endLayerIndex** Optioneel. Een op nul gebaseerde index die aangeeft bij welke laag het kopiëren wordt stopgezet. De reeks lagen die moet worden gekopieerd loopt tot en met `endLayerIndex`. Als u alleen `startLayerIndex` opgeeft, wordt de waarde van `endLayerIndex` standaard ingesteld op de waarde van `startLayerIndex`.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; kopieert de momenteel in de tijdslijn geselecteerde lagen of de lagen in het opgegeven bereik. Er zijn optionele argumenten als u een laag of een reeks lagen wilt opgeven om te kopiëren.

#### Voorbeeld

In het volgende voorbeeld worden de lagen van index 2 tot en met index 7 in de tijdslijn gekopieerd:

```
fl.getDocumentDOM().getTimeline().copyLayers(2, 7);
```

#### Zie ook

[timeline.cutLayers\(\)](#), [timeline.pasteLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

## timeline.copyMotion()

#### Beschikbaarheid

Flash CS3 Professional.

#### Gebruik

```
timeline.copyMotion()
```

#### Parameters

Geen.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt beweging op geselecteerde frames gekopieerd, dit kan zijn van een bewegings-tween of van een frame-voor-frame animatie. U kunt de beweging vervolgens met [timeline.pasteMotion\(\)](#) op andere frames toepassen.

Zie [timeline.copyMotionAsAS3\(\)](#) voor meer informatie over het kopiëren van beweging als tekst (code) die u in een script kunt plakken.

#### Voorbeeld

In het volgende voorbeeld wordt de beweging van één of meer geselecteerde frames gekopieerd:

```
fl.getDocumentDOM().getTimeline().copyMotion();
```

#### Zie ook

[timeline.copyMotionAsAS3\(\)](#), [timeline.pasteMotion\(\)](#)

## timeline.copyMotionAsAS3()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
timeline.copyMotionAsAS3()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt beweging op geselecteerde frames, dit kan zijn van een bewegings-tween of van een frame-voor-frame animatie, als ActionScript 3.0-code gekopieerd naar het klembord. U kunt deze code vervolgens in een script plakken.

Zie [timeline.copyMotion\(\)](#) voor meer informatie over het kopiëren van beweging in een indeling die u op andere frames kunt toepassen.

### Voorbeeld

In het volgende voorbeeld wordt de beweging van één of meer geselecteerde frames als ActionScript 3.0-code naar het klembord gekopieerd:

```
fl.getDocumentDOM().getTimeline().copyMotionAsAS3();
```

### Zie ook

[timeline.copyMotion\(\)](#)

## timeline.createMotionObject()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
timeline.createMotionObject([startFrame [,endFrame]])
```

### Parameters

**startFrame** Bepaalt het eerste frame waarbij bewegingsobjecten worden gemaakt. Als u *startFrame* niet instelt, gebruikt de methode de huidige selectie. Als er geen selectie is, worden alle frames op de huidige afspreekop op alle lagen verwijderd. Deze parameter is optioneel.

**endFrame** Bepaalt bij welk frame het maken van bewegingsobjecten wordt gestopt. Het framebereik loopt tot (maar niet tot en met) *endFrame*. Als u alleen *startFrame* opgeeft, krijgt *endFrame* standaard dezelfde waarde als *startFrame*. Deze parameter is optioneel.



### Geretourneerde waarde

Niets.

### Beschrijving

Methode, hiermee maakt u een nieuw bewegingsobject. De parameters zijn optioneel. Als ze worden opgegeven, dient u de tijdlijnselectie in te stellen op de aangewezen frames voordat u het bewegingsobject maakt.

### Voorbeeld

Het volgende voorbeeld maakt een bewegingsobject op de huidige positie van de afspeelkop op de bovenste laag:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().createMotionObject();
```

Het volgende voorbeeld maakt een bewegingsobject dat begint bij Frame 5 en loopt tot (maar niet tot en met) Frame 15 van de bovenste laag in de actuele scène:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().createMotionObject(5, 15);
```

## timeline.createMotionTween()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.createMotionTween([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die de eerste frame aangeeft waarop een bewegings-tween moet worden gemaakt. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame de bewegings-tween wordt stopgezet. De reeks frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de eigenschap `frame.tweenType` voor elk geselecteerd hoofdframe op de huidige laag ingesteld op `motion` en wordt de inhoud van alle frames zo nodig geconverteerd naar één symboolinstantie. Deze eigenschap is gelijk aan de menuoptie Bewegings-tween maken in het Flash-ontwerpgereedschap.

### Voorbeeld

In het volgende voorbeeld wordt de vorm in het eerste frame tot, maar niet inclusief, Frame 10 geconverteerd naar een grafische symboolinstantie en wordt `frame.tweenType` ingesteld op `motion` (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().createMotionTween(0, 9);
```

## timeline.currentFrame

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.currentFrame
```

### Beschrijving

Eigenschap; de op nul gebaseerde index voor het frame op de huidige locatie van de afspeelkop.

### Voorbeeld

In het volgende voorbeeld wordt de afspeelkop van de huidige tijdlijn ingesteld op Frame 10 (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().currentFrame = 9;
```

In het volgende voorbeeld wordt de waarde van de huidige locatie van de afspeelkop in de variabele `curFrame` opgeslagen:

```
var curFrame = fl.getDocumentDOM().getTimeline().currentFrame;
```

## timeline.currentLayer

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.currentLayer
```

### Beschrijving

Eigenschap; de op nul gebaseerde index voor de momenteel actieve laag. De waarde 0 verwijst naar de toplaag, de waarde 1 verwijst naar de laag eronder, enzovoort.

### Voorbeeld

In het volgende voorbeeld wordt de toplaag actief gemaakt:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
```

In het volgende voorbeeld wordt de index van de momenteel actieve laag in de variabele `curLayer` opgeslagen:

```
var curLayer = fl.getDocumentDOM().getTimeline().currentLayer;
```

## timeline.cutFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.cutFrames([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het begin aangeeft van een reeks frames die moet worden geknipt. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame het knippen wordt stopgezet. De reeks frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode: hiermee wordt een reeks frames op de huidige laag van de tijdlijn geknipt en naar het klembord gekopieerd.

### Voorbeeld

In het volgende voorbeeld worden de geselecteerde frames uit de tijdlijn geknipt en op het klembord opgeslagen:

```
fl.getDocumentDOM().getTimeline().cutFrames();
```

In het volgende voorbeeld wordt Frame 2 tot, maar niet inclusief, Frame 10 van de tijdlijn geknipt en opgeslagen naar klembord (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().cutFrames(1, 9);
```

In het volgende voorbeeld wordt Frame 5 uit de tijdlijn geknipt en op het klembord opgeslagen:

```
fl.getDocumentDOM().getTimeline().cutFrames(4);
```

## timeline.cutLayers()

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
timeline.cutLayers([startLayerIndex [, endLayerIndex]])
```

### Parameters

**startLayerIndex** Optioneel. Een op nul gebaseerde index die het begin aangeeft van de reeks lagen die moet worden geknipt. Als u *startLayerIndex* niet instelt, gebruikt de methode de huidige selectie.

**endLayerIndex** Optioneel. Een op nul gebaseerde index die aangeeft bij welke laag het knippen wordt stopgezet. De reeks lagen die moet worden geknipt, loopt tot en met *endLayerIndex*. Als u alleen *startLayerIndex* opgeeft, wordt de waarde van *endLayerIndex* standaard ingesteld op de waarde van *startLayerIndex*.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; knipt de momenteel in de tijdlijn geselecteerde lagen of de lagen in het opgegeven bereik. Er zijn optionele argumenten als u een laag of een reeks lagen wilt opgeven om te knippen.

### Voorbeeld

In het volgende voorbeeld worden de lagen van index 2 tot en met index 7 in de tijdlijn geknipt:

```
fl.getDocumentDOM().getTimeline().cutLayers(2, 7);
```

### Zie ook

[timeline.copyLayers\(\)](#), [timeline.pasteLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

## timeline.deleteLayer()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.deleteLayer([index])
```

### Parameters

**index** Een op nul gebaseerde index die de te verwijderen laag aangeeft. Als er maar één laag in de tijdlijn staat, heeft deze methode geen effect. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een laag verwijderd. Als de laag een map is, worden alle lagen in de map verwijderd. Als u geen laagindex opgeeft, worden de geselecteerde lagen verwijderd.

### Voorbeeld

In het volgende voorbeeld wordt de tweede laag van boven verwijderd:

```
fl.getDocumentDOM().getTimeline().deleteLayer(1);
```

In het volgende voorbeeld worden de momenteel geselecteerde lagen verwijderd:

```
fl.getDocumentDOM().getTimeline().deleteLayer();
```

## timeline.duplicateLayers()

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
timeline.duplicateLayers([startLayerIndex [, endLayerIndex]])
```

### Parameters

**startLayerIndex** Optioneel. Een op nul gebaseerde index die het begin aangeeft van de reeks lagen die moet worden gekopieerd. Deze index geeft ook de laag aan waarop de lagen op het klembord worden geplakt. Als u `startLayerIndex` niet instelt, gebruikt de methode de laag die momenteel is geselecteerd.

**endLayerIndex** Optioneel. Een op nul gebaseerde index die aangeeft bij welke laag het kopiëren wordt stopgezet. De reeks lagen die moet worden gekopieerd loopt tot en met `endLayerIndex`. Als u alleen `startLayerIndex` opgeeft, wordt de waarde van `endLayerIndex` standaard ingesteld op de waarde van `startLayerIndex`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; dupliceert de momenteel in de tijdlijn geselecteerde lagen of de lagen in het opgegeven bereik. Er zijn optionele argumenten als u een laag of een reeks lagen wilt opgeven om te dupliceren.

### Voorbeeld

In het volgende voorbeeld wordt de laag gedupliceerd die momenteel in de tijdlijn is geselecteerd:

```
fl.getDocumentDOM().getTimeline().duplicateLayers();
```

In het volgende voorbeeld worden de lagen van index 2 tot index 7 op laagindex 2 gedupliceerd:

```
fl.getDocumentDOM().getTimeline().duplicatedLayers(2,7);
```

### Zie ook

[timeline.copyLayers\(\)](#), [timeline.cutLayers\(\)](#), [timeline.pasteLayers\(\)](#)

## timeline.expandFolder()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.expandFolder(bExpand [, bRecurseNestedParents [, index]])
```

### Parameters

**bExpand** Een Booleaanse waarde: bij `true` wordt de map uitgevouwen; bij `false` wordt de map samengevouwen.

**bRecurseNestedParents** Een Booleaanse waarde: bij `true` worden alle lagen in de opgegeven map geopend of gesloten, afhankelijk van de parameter *bExpand*. Deze parameter is optioneel.

**index** Een op nul gebaseerde index van de open of samen te vouwen map. Gebruik -1 om de methode op alle lagen toe te passen (u moet daarvoor ook *bRecurseNestedParents* op `true` instellen). Deze eigenschap is gelijk aan de menuoptie Alles uitvouwen/Alles samenvouwen in het Flash-ontwerpgereedschap. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden de opgegeven map of mappen uit- of samengevouwen. Als u geen laag opgeeft, wordt de methode uitgevoerd op de huidige laag.

### Voorbeeld

In de onderstaande voorbeelden wordt deze mapstructuur gebruikt:

```
Folder 1 ***
--layer 7
--Folder 2 ****
---Layer 5
```

In het volgende voorbeeld wordt alleen Folder 1 uitgevouwen:

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;
fl.getDocumentDOM().getTimeline().expandFolder(true);
```

In het volgende voorbeeld wordt alleen Folder 1 uitgevouwen (ervan uitgaande dat Folder 2 samen met Folder 1 werd samengevouwen, anders wordt Folder 2 ook uitgevouwen):

```
fl.getDocumentDOM().getTimeline().expandFolder(true, false, 0);
```

In het volgende voorbeeld worden alle mappen in de huidige tijdlijn samengevouwen:

```
fl.getDocumentDOM().getTimeline().expandFolder(false, true, -1);
```

## timeline.findLayerIndex()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.findLayerIndex(name)
```

### Parameters

**name** Een tekenreeks die de naam van de gezochte laag aangeeft.

### Geretourneerde waarde

Een array van indexwaarden voor de opgegeven laag. Als de opgegeven laag niet wordt gevonden, wordt `undefined` geretourneerd.

### Beschrijving

Methode; hiermee wordt gezocht naar een array van indexen voor de lagen met de opgegeven naam. De laagindex is vlak, mappen worden dus als deel van de hoofdindex beschouwd.

### Voorbeeld

In het volgende voorbeeld worden de indexwaarden van alle lagen met de naam `Layer 7` weergegeven in het deelvenster Uitvoer:

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 7");
fl.trace(layerIndex);
```

In het volgende voorbeeld wordt getoond hoe de waarden die door deze methode worden geretourneerd, worden doorgegeven aan `timeline.setSelectedLayers()`:

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 1");  
fl.getDocumentDOM().getTimeline().setSelectedLayers(layerIndex[0], true);
```

## timeline.frameCount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.frameCount
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat het aantal frames in de langste laag van deze tijdlijn vertegenwoordigt.

### Voorbeeld

In het volgende voorbeeld wordt het aantal frames in de langste laag van het huidige document opgeslagen in de variabele `countNum`:

```
var countNum = fl.getDocumentDOM().getTimeline().frameCount;
```

## timeline.getFrameProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.getFrameProperty(property [, startFrameIndex [, endFrameIndex]])
```

### Parameters

**eigenschap** Een tekenreeks die de naam aangeeft van de eigenschap waarvoor de waarde moet worden opgehaald. Een volledige lijst met eigenschappen vindt u in het Overzicht van eigenschappen voor het [Object Frame](#).

**startFrameIndex** Een op nul gebaseerde index die het nummer aangeeft van het eerste frame waarvoor de waarde moet worden opgehaald. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die het einde aangeeft van de reeks frames die moet worden geselecteerd. De reeks loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Een waarde voor de opgegeven eigenschap, of `undefined` wanneer niet alle geselecteerde frames dezelfde eigenschapswaarde hebben.

### Beschrijving

Methode; hiermee wordt de waarde van de opgegeven eigenschap voor de geselecteerde frames opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt de naam van het eerste frame in de toplaag van het huidige document opgehaald en wordt deze in het deelvenster Uitvoer weergegeven:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 0, true);  
var frameName = fl.getDocumentDOM().getTimeline().getFrameProperty("name");  
fl.trace(frameName);
```

## timeline.getGuidelines()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
timeline.getGuidelines()
```

### Parameters

Geen.

### Geretourneerde waarde

Een XML-tekenreeks.

### Beschrijving

Methode; hiermee wordt een XML-tekenreeks geretourneerd die de huidige posities van de horizontale en verticale hulplijnen voor een tijdlijn vertegenwoordigt (Weergave > Hulplijnen > Hulplijnen weergeven). Gebruik [timeline.setGuidelines\(\)](#) om deze hulplijnen op een tijdlijn toe te passen.

### Voorbeeld

Wanneer er op de eerste tijdlijn enkele hulplijnen staan, worden deze in het volgende voorbeeld als XML-tekenreeks weergegeven in het deelvenster Uitvoer:

```
var currentTimeline = fl.getDocumentDOM().timelines[0];  
fl.trace(currentTimeline.getGuidelines());
```

## timeline.getLayerProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.getLayerProperty(property)
```



### Parameters

**eigenschap** Een tekenreeks die de naam aangeeft van de eigenschap waarvoor u de waarde wilt ophalen. Een lijst met eigenschappen vindt u in het Overzicht van eigenschappen voor het [Object Frame](#).

### Geretourneerde waarde

De waarde van de opgegeven eigenschap. Flash bepaalt het type op basis van de eigenschappen van de laag. Als niet alle opgegeven lagen dezelfde eigenschapswaarde hebben, wordt `undefined` geretourneerd.

### Beschrijving

Methode; hiermee wordt de waarde van de opgegeven eigenschap voor de geselecteerde lagen opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de toplaag in het huidige document opgehaald en wordt deze in het deelvenster Uitvoer weergegeven:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
var layerName = fl.getDocumentDOM().getTimeline().getLayerProperty("name");  
fl.trace(layerName);
```

## timeline.getSelectedFrames()

### Beschikbaarheid

Flash MX 2004.

### Parameters

Geen.

### Geretourneerde waarde

Een array met 3 gehele getallen  $n$ , waarbij  $n$  het aantal geselecteerde regio's voorstelt. Het eerste gehele getal in elke groep is de laagindex, het tweede is het eerste frame aan het begin van de selectie, en het derde geeft het eindframe van het selectiebereik aan. Het eindframe is niet opgenomen in de selectie.

### Beschrijving

Methode; hiermee worden de momenteel geselecteerde frames in een array opgevraagd.

### Voorbeeld

Omdat de toplaag ook de huidige laag is, wordt in het volgende voorbeeld `0, 5, 10, 0, 20, 25` weergegeven in het deelvenster Uitvoer:

```
var timeline = fl.getDocumentDOM().getTimeline();  
timeline.setSelectedFrames(5, 10);  
timeline.setSelectedFrames(20, 25, false);  
var theSelectedFrames = timeline.getSelectedFrames();  
fl.trace(theSelectedFrames);
```

### Zie ook

[timeline.setSelectedFrames\(\)](#)

## timeline.getSelectedLayers()

### Beschikbaarheid

Flash MX 2004.

### Parameters

Geen.

### Geretourneerde waarde

Een array van de op nul gebaseerde indexwaarden van de geselecteerde lagen.

### Beschrijving

Methode; hiermee worden de nul gebaseerde indexwaarden van de momenteel geselecteerde lagen opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt 1, 0 weergegeven in het deelvenster Uitvoer:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);  
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);  
var layerArray = fl.getDocumentDOM().getTimeline().getSelectedLayers();  
fl.trace(layerArray);
```

### Zie ook

[timeline.setSelectedLayers\(\)](#)

## timeline.insertBlankKeyframe()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.insertBlankKeyframe([frameNumIndex])
```

### Parameters

**frameNumIndex** Een op nul gebaseerde index die aangeeft bij welk frame het hoofdfame moet worden ingevoegd. Als u *frameNumIndex* niet instelt, gebruikt de methode het huidige framenummer van de afspeelkop. Deze parameter is optioneel.

Als het opgegeven of geselecteerde frame een regulier frame is, wordt het hoofdfame bij dat frame ingevoegd. U hebt bijvoorbeeld een reeks van 10 frames met de nummers 1-10 en selecteert Frame 5. Deze methode maakt dan van Frame 5 een leeg hoofdfame, en de lengte van de reeks blijft 10 frames. Als Frame 5 wordt geselecteerd en dit een hoofdfame is met een regulier frame er naast, dan voegt deze methode bij Frame 6 een leeg hoofdfame in. Als Frame 5 en het frame ernaast als hoofdfame zijn, dan wordt geen hoofdfame ingevoegd maar wordt de afspeelkop verplaatst naar Frame 6.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt in de opgegeven frame-index een leeg hoofdfame ingevoegd. Als de index niet is opgegeven, voegt de methode het lege hoofdfame aan de hand van de afspeelkop of de locatie van de selectie in. Zie ook `timeline.insertKeyframe()`.

### Voorbeeld

In het volgende voorbeeld wordt bij Frame 20 een leeg hoofdfame ingevoegd (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe(19);
```

In het volgende voorbeeld wordt bij het momenteel geselecteerde frame (of op de locatie van de afspeelkop als er geen frame is geselecteerd) een leeg hoofdfame ingevoegd:

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe();
```

## timeline.insertFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.insertFrames([numFrames [, bAllLayers [, frameNumIndex]])
```

### Parameters

**numFrames** Een geheel getal dat het aantal in te voegen frames aangeeft. Als u deze parameter niet instelt, voegt de methode frames in bij de huidige selectie in de huidige laag. Deze parameter is optioneel.

**bAllLayers** Een Booleaanse waarde: bij `true` (standaard) voegt de methode het in de parameter `numFrames` opgegeven aantal frames in alle lagen in; bij `false` voegt de methode frames in de huidige laag in. Deze parameter is optioneel.

**frameNumIndex** Een op nul gebaseerde index die aangeeft bij welk frame een nieuw frame moet worden ingevoegd. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het opgegeven aantal frames bij de opgegeven index ingevoegd.

Als er geen parameters zijn opgegeven, werkt deze methode als volgt:

- Als er één of meerdere frames zijn geselecteerd, voegt de methode het geselecteerde aantal frames op de locatie van het eerst geselecteerde frame in de huidige laag in. Dus als frame 6 tot en met 10 zijn geselecteerd (in totaal vijf frames), dan voegt de methode bij Frame 6 vijf frames in de laag met geselecteerde frames in.
- Als er geen frames zijn geselecteerd, voegt de methode bij het huidige frame op alle lagen één frame in.

Als er parameters zijn opgegeven, werkt deze methode als volgt:

- Als alleen `numFrames` is opgegeven, wordt het opgegeven aantal frames bij het huidige frame op de huidige laag ingevoegd.

- Als *numFrames* is opgegeven en *bAllLayers* is ingesteld op `true`, wordt het opgegeven aantal frames bij het huidige frame op alle lagen ingevoegd.
- Als alle drie parameters zijn opgegeven, wordt het opgegeven aantal frames bij de opgegeven index (*frameIndex*) ingevoegd; de voor *bAllLayers* doorgegeven waarde bepaalt of de frames alleen aan de huidige laag of aan alle lagen worden toegevoegd.

Als het opgegeven of geselecteerde frame een regulier frame is, wordt het frame bij dat frame ingevoegd. U hebt bijvoorbeeld een reeks van 10 frames met de nummers 1-10 en selecteert Frame 5 (of geeft de waarde 4 door voor *frameIndex*). Deze methode voegt dan bij Frame 5 een frame in en de lengte van de reeks wordt 11 frames. Als Frame 5 wordt geselecteerd en dit een hoofdfraam is, dan voegt deze methode bij Frame 6 een frame in, ook als het frame ernaast een hoofdfraam is.

### Voorbeeld

In het volgende voorbeeld worden één of meer frames (afhankelijk van de selectie) bij de huidige selectie in de huidige laag ingevoegd:

```
fl.getDocumentDOM().getTimeline().insertFrames();
```

In het volgende voorbeeld worden bij het huidige frame vijf frames in alle lagen ingevoegd:

```
fl.getDocumentDOM().getTimeline().insertFrames(5);
```

**Opmerking:** Als er meerdere lagen zijn die frames bevatten, en u met de vorige opdracht in één laag een frame selecteert, dan worden de frames alleen ingevoegd in de geselecteerde laag. Als er meerdere lagen zijn waarin geen frames werden geselecteerd, dan worden de frames in alle lagen ingevoegd.

In het volgende voorbeeld worden drie frames alleen in de huidige laag ingevoegd:

```
fl.getDocumentDOM().getTimeline().insertFrames(3, false);
```

In het volgende voorbeeld worden in alle lagen vier frames ingevoegd, te beginnen bij het eerste frame:

```
fl.getDocumentDOM().getTimeline().insertFrames(4, true, 0);
```

## timeline.insertKeyframe()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.insertKeyframe([frameNumIndex])
```

### Parameters

**frameNumIndex** Een op nul gebaseerde index die aangeeft bij welke frame-index het hoofdfraam in de huidige laag moet worden ingevoegd. Als u *frameNumIndex* niet instelt, gebruikt de methode het framenummer van de huidige afspeelkop of het geselecteerde frame. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een hoofdfraam ingevoegd bij het opgegeven frame. Als u de parameter niet instelt, dan voegt de methode aan de hand van de afspeelkop of de locatie van de selectie een hoofdfraam in.

Deze methode werkt op dezelfde manier als `timeline.insertBlankKeyframe()`. Het verschil is dat het ingevoegde hoofdfraam de inhoud bevat van het frame dat het heeft geconverteerd (het frame is dus niet leeg).

### Voorbeeld

In het volgende voorbeeld wordt een hoofdfraam bij de afspeelkop of geselecteerde locatie ingevoegd:

```
fl.getDocumentDOM().getTimeline().insertKeyframe();
```

In het volgende voorbeeld wordt bij Frame 10 van de tweede laag een hoofdfraam ingevoegd (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;  
fl.getDocumentDOM().getTimeline().insertKeyframe(9);
```

## timeline.layerCount

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.layerCount
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat het aantal lagen in de opgegeven tijdlijn vertegenwoordigt.

### Voorbeeld

In het volgende voorbeeld wordt het aantal lagen in de huidige scène opgeslagen in de variabele `NumLayer`:

```
var NumLayer = fl.getDocumentDOM().getTimeline().layerCount;
```

## timeline.layers

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.layers
```

### Beschrijving

Alleen-lezen eigenschap; een array van laagobjecten.

### Voorbeeld

In het volgende voorbeeld wordt de array van laagobjecten in het huidige document opgeslagen in de variabele `currentLayers`:

```
var currentLayers = fl.getDocumentDOM().getTimeline().layers;
```

## timeline.libraryItem

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
timeline.libraryItem
```

### Beschrijving

Alleen-lezen eigenschap; als de eigenschap `libraryItem` van de tijdlijn null is, maakt de tijdlijn deel uit van een scène. Als de eigenschap niet null is, kunt u deze op dezelfde manier behandelen als een `LibraryItem`-object.

### Voorbeeld

In het volgende voorbeeld wordt de naam van `LibraryItem` geretourneerd als de waarde van `libraryItem` niet null is. Als `libraryItem` null is, wordt de naam van de scène uitgevoerd:

```
var item = fl.getDocumentDOM().getTimeline().libraryItem;
if (item)
    fl.trace("libraryItem name: " + item.name);
else
    fl.trace("scene name: " + fl.getDocumentDOM().getTimeline().name);
```

## timeline.name

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.name
```

### Beschrijving

Eigenschap; een tekenreeks die de naam van de huidige tijdlijn aangeeft. Dit is de naam van de huidige scène, het huidige scherm (dia of document) of van het symbool dat wordt bewerkt.

### Voorbeeld

In het volgende voorbeeld wordt de naam van de eerste scène opgehaald:

```
var sceneName = fl.getDocumentDOM().timelines[0].name;
```

In het volgende voorbeeld wordt de naam van de eerste scène op `FirstScene` ingesteld:

```
fl.getDocumentDOM().timelines[0].name = "FirstScene";
```

## timeline.pasteFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.pasteFrames([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die het begin aangeeft van een reeks frames die moet worden geplakt. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame het plakken wordt stopgezet. De methode plakt tot, maar niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode: hiermee wordt de reeks frames van het klembord in de opgegeven frames geplakt.

### Voorbeeld

In het volgende voorbeeld worden de frames op het klembord in het momenteel geselecteerde frame of op de locatie van de afspeelkop geplakt:

```
fl.getDocumentDOM().getTimeline().pasteFrames();
```

In het volgende voorbeeld worden de frames op het klembord vanaf Frame 2 tot, maar niet inclusief, Frame 10 geplakt (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().pasteFrames(1, 9);
```

In het volgende voorbeeld worden de frames op het klembord vanaf Frame 5 geplakt:

```
fl.getDocumentDOM().getTimeline().pasteFrames(4);
```

## timeline.pasteLayers()

### Beschikbaarheid

Flash CS5.5 Professional.

### Gebruik

```
timeline.pasteLayers([layerIndex])
```

### Parameters

**layerIndex** Optioneel. Een op nul gebaseerde index die de laag aangeeft waarop de lagen op het klembord worden geplakt. Als u *layerIndex* niet instelt, gebruikt de methode de huidige selectie.

### Geretourneerde waarde

Geheel getal dat naar de laagste laagindex van de geplakte lagen verwijst.

### Beschrijving

Methode; plak lagen die eerder zijn geknipt of gekopieerd boven de momenteel geselecteerde laag of boven de opgegeven laagindex. Wanneer de opgegeven laag een maplaag is, worden de lagen in de map geplakt. Retourneert de laagste laagindex van de geplakte lagen. Deze handeling beïnvloedt het klembord van het systeem niet.

### Voorbeeld

In het volgende voorbeeld worden de lagen van het lagenklembord boven de momenteel geselecteerde laag in de tijdlijn geplakt:

```
fl.getDocumentDOM().getTimeline().pasteLayers();
```

In het volgende voorbeeld worden de lagen van het klembord op laagindex 2 geplakt:

```
fl.getDocumentDOM().getTimeline().pasteLayers(2);
```

### Zie ook

[timeline.cutLayers\(\)](#), [timeline.copyLayers\(\)](#), [timeline.duplicateLayers\(\)](#)

## timeline.pasteMotion()

### Beschikbaarheid

Flash CS3 Professional.

### Gebruik

```
timeline.pasteMotion()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt hiermee de reeks bewegingsframes die is opgehaald door [timeline.copyMotion\(\)](#), in de tijdlijn geplakt. Bestaande frames worden indien nodig verplaatst (naar rechts), zodat er plaats vrijkomt om de frames te plakken.

### Voorbeeld

In het volgende voorbeeld wordt de beweging op het klembord in het momenteel geselecteerde frame of op de locatie van de afspeelkop geplakt, waarbij dat frame rechts van de geplakte frames komt te staan:

```
fl.getDocumentDOM().getTimeline().pasteMotion();
```

### Zie ook

[timeline.copyMotion\(\)](#)



## timeline.removeFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.removeFrames([startFrameIndex [,endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die aangeeft welk frame als eerste moet worden verwijderd. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Als er geen selectie is, worden alle frames aan de huidige afspreekop op alle lagen verwijderd. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame niet meer wordt verwijderd; de reeks frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het frame verwijderd.

### Voorbeeld

In het volgende voorbeeld wordt Frame 5 tot, maar niet inclusief, Frame 10 van de toplaag in de huidige scène verwijderd (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeFrames(4, 9);
```

In het volgende voorbeeld wordt Frame 8 op de toplaag in de huidige scène verwijderd:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeFrames(7);
```

## timeline.removeMotionObject()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
timeline.removeMotionObject([startFrame [,endFrame]])
```

### Parameters

**startFrame** Bepaalt het eerste frame waarbij wordt begonnen met het verwijderen van bewegingsobjecten. Als u *startFrame* niet instelt, gebruikt de methode de huidige selectie. Als er geen selectie is, worden alle frames op de huidige afspreekop op alle lagen verwijderd. Deze parameter is optioneel.

**endFrame** Bepaalt bij welk frame het verwijderen van bewegingsobjecten wordt gestopt. Het framebereik loopt tot (maar niet tot en met) *endFrame*. Als u alleen *startFrame* opgeeft, krijgt *endFrame* standaard dezelfde waarde als *startFrame*. Deze parameter is optioneel.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode, hiermee verwijdert u het bewegingsobject en zet u het frame of de frames weer terug in statische frames. De parameters zijn optioneel. Als ze worden opgegeven, dient u de tijdlijnselectie in te stellen op de aangewezen frames voordat u het bewegingsobject verwijdert.

#### Voorbeeld

Het volgende voorbeeld verwijdert alle bewegingsobjecten en zet de frames weer om in statische frames op de huidige positie van de afspeelkop op de bovenste laag:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeMotionObject();
```

Het volgende voorbeeld verwijdert de bewegingsobjecten tussen Frame 5 en Frame 15 (exclusief Frame 15) van de bovenste laag in de huidige scène:

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;  
fl.getDocumentDOM().getTimeline().removeMotionObject(5, 15);
```

#### Zie ook

`"timeline.createMotionObject\(\)"` op pagina 482

## timeline.reorderLayer()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
timeline.reorderLayer(layerToMove, layerToPutItBy [, bAddBefore])
```

#### Parameters

**layerToMove** Een op nul gebaseerd index die aangeeft welke laag moet worden verplaatst.

**layerToPutItBy** Een op nul gebaseerde index die aangeeft naast welke laag u de laag wilt plaatsen. Als u bijvoorbeeld bij *layerToMove* 1 en bij *layerToPutItBy* 0 opgeeft, wordt de tweede laag naast de eerste laag geplaatst.

**bAddBefore** Hiermee wordt aangegeven aan of de laag voor of na *layerToPutItBy* komt te staan. Bij *false* komt de laag na *layerToPutItBy* te staan. De standaardwaarde is *true*. Deze parameter is optioneel.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de eerste opgegeven laag voor of achter de tweede opgegeven laag geplaatst.

### Voorbeeld

In het volgende voorbeeld wordt de laag bij index 2 naar boven verplaatst (boven de laag bij index 0):

```
fl.getDocumentDOM().getTimeline().reorderLayer(2, 0);
```

In het volgende voorbeeld wordt de laag bij index 3 achter de laag bij index 5 geplaatst:

```
fl.getDocumentDOM().getTimeline().reorderLayer(3, 5, false);
```

## timeline.reverseFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.reverseFrames([startFrameIndex [, endFrameIndex]])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die aangeeft welk frame als eerste moet worden omgekeerd. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame niet meer wordt omgekeerd; de reeks frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een reeks frames omgekeerd.

### Voorbeeld

In het volgende voorbeeld worden de posities van de momenteel geselecteerde frames omgekeerd:

```
fl.getDocumentDOM().getTimeline().reverseFrames();
```

In het volgende voorbeeld worden de frames vanaf Frame 10 tot, maar niet inclusief, Frame 15 omgekeerd (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().reverseFrames(9, 14);
```

## timeline.selectAllFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.selectAllFrames()
```

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden alle frames in de huidige tijdlijn geselecteerd.

### Voorbeeld

In het volgende voorbeeld worden alle frames in de huidige tijdlijn geselecteerd.

```
fl.getDocumentDOM().getTimeline().selectAllFrames();
```

## timeline.setFrameProperty()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.setFrameProperty(property, value [, startFrameIndex [, endFrameIndex]])
```

### Parameters

**eigenschap** Een tekenreeks die de naam aangeeft van de eigenschap die u wilt wijzigen. Zie de overzichtstabel met eigenschappen van het [Object Frame](#) voor een lijst eigenschappen en waarden.

U kunt met deze methode geen waarden instellen voor alleen-lezen eigenschappen, zoals [frame.duration](#) en [frame.elements](#).

**value** Hiermee wordt de waarde opgegeven waarop u de eigenschap wilt instellen. Zie de overzichtstabel met eigenschappen van het [Object Frame](#) voor geldige waarden en typen.

**startFrameIndex** Een op nul gebaseerde index die het eerste frame aangeeft dat moet worden gewijzigd. Als u *startFrameIndex* niet instelt, gebruikt de methode de huidige selectie. Deze parameter is optioneel.

**endFrameIndex** Een op nul gebaseerde index die aangeeft bij welk frame het wijzigen wordt stopgezet. De reeks frames loopt tot, maar is niet inclusief, *endFrameIndex*. Als u alleen *startFrameIndex* invoert en geen *endFrameIndex*, krijgt *endFrameIndex* dezelfde waarde als *startFrameIndex*. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de eigenschap van het object Frame ingesteld voor de geselecteerde frames.

### Voorbeeld

In het volgende voorbeeld wordt de ActionScript-opdracht `stop()` toegekend aan het eerste frame van de toplaat in het huidige document:

**Object Timeline**

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().setSelectedFrames(0,0,true);
fl.getDocumentDOM().getTimeline().setFrameProperty("actionScript", "stop();");
```

In het volgende voorbeeld wordt een bewegings-tween ingesteld van Frame 2 tot, maar niet inclusief, Frame 5 (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
var doc = fl.getDocumentDOM();
doc.getTimeline().setFrameProperty("tweenType", "motion", 1, 4);
```

## timeline.setGuidelines()

**Beschikbaarheid**

Flash CS4 Professional.

**Gebruik**

```
timeline.setGuidelines(xmlString)
```

**Parameters**

**xmlString** Een XML-tekenreeks met informatie over de hulplijnen die u wilt toepassen.

**Geretourneerde waarde**

Een Booleaanse waarde: `true` als de hulplijnen zijn toegepast, anders `false`.

**Beschrijving**

Methode: hiermee worden de hulplijnen voor de tijdlijn (Weergave > Hulplijnen > Hulplijnen weergeven) vervangen door de informatie in *xmlString*. Gebruik `timeline.getGuidelines()` om een XML-tekenreeks op te halen die u kunt doorgeven aan deze methode.

Mogelijk moet u de hulplijnen eerst verbergen en weergeven voordat de ingestelde hulplijnen zichtbaar zijn.

**Voorbeeld**

In het volgende voorbeeld worden de hulplijnen van een FLA-bestand toegepast op een ander FLA-bestand:

```
var doc0 = fl.documents[0];
var guides0 = doc0.timelines[0].getGuidelines();
var doc1 = fl.documents[1];
doc1.timelines[0].setGuidelines(guides0);
```

## timeline.setLayerProperty()

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
timeline.setLayerProperty(property, value [, layersToChange])
```

### Parameters

**eigenschap** Een tekenreeks die de in te stellen eigenschap aangeeft. Zie “Object Layer” op pagina 327 voor een lijst met eigenschappen.

**value** De waarde waarop u de eigenschap wilt instellen. Gebruik hetzelfde type waarde dat u zou gebruiken om de eigenschap in het laagobject in te stellen.

**layersToChange** Een tekenreeks die aangeeft welke lagen moeten worden gewijzigd. Acceptabele waarden zijn "selected", "all" en "others". Als u deze parameter niet instelt, wordt de standaardwaarde "selected" gebruikt. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de opgegeven eigenschap voor alle geselecteerde lagen ingesteld op een opgegeven waarde.

### Voorbeeld

In het volgende voorbeeld worden een of meer geselecteerde lagen onzichtbaar gemaakt:

```
fl.getDocumentDOM().getTimeline().setLayerProperty("visible", false);
```

In het volgende voorbeeld krijgen een of meer geselecteerde lagen de naam selLayer:

```
fl.getDocumentDOM().getTimeline().setLayerProperty("name", "selLayer");
```

## timeline.setSelectedFrames()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.setSelectedFrames(startFrameIndex, endFrameIndex [, bReplaceCurrentSelection])  
timeline.setSelectedFrames(selectionList [, bReplaceCurrentSelection])
```

### Parameters

**startFrameIndex** Een op nul gebaseerde index die aangeeft welk frame als eerste moet worden ingesteld.

**endFrameIndex** Een op nul gebaseerde index die het einde van de selectie aangeeft. *endFrameIndex* is het frame na het laatste frame in de te selecteren reeks.

**bReplaceCurrentSelection** Een Booleaanse waarde: bij `true` wordt de huidige selectie opgeheven voordat de opgegeven frames worden geselecteerd. De standaardwaarde is `true`.

**selectionList** Een array van drie gehele getallen, zoals geretourneerd door `timeline.getSelectedFrames()`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een reeks frames in de huidige laag geselecteerd of worden de geselecteerde frames ingesteld in de selectiearray die wordt doorgegeven aan deze methode.

### Voorbeeld

In het onderstaande voorbeeld worden twee manieren getoond om Frame 1 tot, maar niet inclusief, Frame 10 op de bovenste laag te selecteren, en om vervolgens Frame 12 tot, maar niet inclusief, Frame 15 op dezelfde laag toe te voegen aan de huidige selectie (houd er rekening mee dat indexwaarden niet gelijk zijn aan framenummers):

```
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 9);  
fl.getDocumentDOM().getTimeline().setSelectedFrames(11, 14, false);  
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 0, 9]);  
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 11, 14], false);
```

In het volgende voorbeeld wordt eerst de array met geselecteerde frames opgeslagen in de variabele `savedSelectionList`. De array wordt later in de code opnieuw gebruikt om die frames weer te selecteren nadat de selectie als gevolg van een opdracht of gebruikersinteractie is gewijzigd:

```
var savedSelectionList = fl.getDocumentDOM().getTimeline().getSelectedFrames();  
// Do something that changes the selection.  
fl.getDocumentDOM().getTimeline().setSelectedFrames(savedSelectionList);
```

### Zie ook

[timeline.getSelectedFrames\(\)](#)

## timeline.setSelectedLayers()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
timeline.setSelectedLayers(index [, bReplaceCurrentSelection])
```

### Parameters

**index** Een op nul gebaseerde index voor de te selecteren laag.

**bReplaceCurrentSelection** Een Booleaanse waarde: bij `true` wordt de huidige selectie vervangen; bij `false` wordt de huidige selectie uitgebreid. De standaardwaarde is `true`. Deze parameter is optioneel.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee ingesteld welke laag moet worden geselecteerd en wordt de opgegeven laag de huidige laag. Door een laag te selecteren worden ook alle frames in de laag geselecteerd.

### Voorbeeld

In het volgende voorbeeld wordt de toplaag geselecteerd:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
```

In het volgende voorbeeld wordt de volgende laag aan de selectie toegevoegd:

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
```

#### Zie ook

[timeline.getSelectedLayers\(\)](#)

## timeline.showLayerMasking()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
timeline.showLayerMasking([layer])
```

#### Parameters

**laag** Een op nul gebaseerde index van een masker of gemaskeerde laag die de maskering tijdens het ontwerp weergeeft. Deze parameter is optioneel.

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt de maskering van de laag tijdens het ontwerp weergegeven door het masker en de gemaskeerde lagen te vergrendelen. Als er geen laag is opgegeven, wordt de huidige laag gebruikt. Als u deze methode gebruikt op een laag die niet van het type Mask of Masked is, dan verschijnt er een foutmelding in het deelvenster Uitvoer.

#### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat de maskering van de eerste laag tijdens het ontwerp moet worden weergegeven.

```
fl.getDocumentDOM().getTimeline().showLayerMasking(0);
```

## timeline.startPlayback()

#### Beschikbaarheid

Flash Professional CS5.

#### Gebruik

```
timeline.startPlayback()
```

#### Geretourneerde waarde

Niets.



### Beschrijving

Methode; hiermee begint de tijdlijn automatisch met afspelen als deze op dat moment wordt afgespeeld. Deze methode kan worden gebruikt bij SWF-deelvensters om het afspelen van de tijdlijn te beheren in de ontwerpomgeving.

### Voorbeeld

In het volgende voorbeeld wordt het afspelen van de tijdlijn gestart.

```
fl.getDocumentDOM().getTimeline().startPlayback();
```

## timeline.stopPlayback()

### Beschikbaarheid

Flash Professional CS5.

### Gebruik

```
timeline.stopPlayback();
```

### Geretourneerde waarde

Niets.

### Beschrijving

Methode, hiermee wordt het automatisch afspelen van de tijdlijn gestopt als deze op het moment wordt afgespeeld. Deze methode kan worden gebruikt bij SWF-deelvensters om het afspelen van de tijdlijn te beheren in de ontwerpomgeving.

### Voorbeeld

In het volgende voorbeeld wordt het afspelen van de tijdlijn gestopt.

```
fl.getDocumentDOM().getTimeline().stopPlayback();
```

# Hoofdstuk 46: Object ToolObj

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Een object ToolObj vertegenwoordigt een individueel gereedschap in het deelvenster Gereedschappen. Gebruik eigenschappen van het [Object Tools](#) om een object ToolObj te benaderen: de array `tools.toolObjs` of `tools.activeTool`.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object ToolObj:

**Opmerking:** De volgende methoden worden alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

Methode	Beschrijving
<code>toolObj.enablePIControl()</code>	Hiermee wordt het opgegeven besturingselement in Eigenschapcontrole in- of uitgeschakeld. Wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.
<code>toolObj.setIcon()</code>	Hiermee wordt een PNG-bestand geïdentificeerd dat als een gereedschappictogram moet worden gebruikt in het deelvenster Gereedschappen van Flash.
<code>toolObj.setMenuString()</code>	Hiermee wordt de tekenreeks ingesteld die in het pop-upmenu verschijnt als de naam voor het gereedschap.
<code>toolObj.setOptionsFile()</code>	Hiermee wordt een XML-bestand gekoppeld aan het gereedschap.
<code>toolObj.setPI()</code>	Hiermee wordt een specifieke Eigenschapcontrole ingesteld die moet worden gebruikt wanneer het gereedschap wordt geactiveerd.
<code>toolObj.setToolName()</code>	Hiermee wordt een naam aan het gereedschap toegewezen voor de configuratie van het deelvenster Gereedschappen.
<code>toolObj.setToolTip()</code>	Hiermee wordt de knopinfo ingesteld die wordt weergegeven wanneer de muis boven het gereedschappictogram beweegt.
<code>toolObj.showPIControl()</code>	Hiermee wordt een besturingselement in Eigenschapcontrole weergegeven of verborgen.
<code>toolObj.showTransformHandles()</code>	Wordt aangeroepen in de methode <code>configureTool()</code> van het JavaScript-bestand van een uitbreidbaar gereedschap om aan te geven dat de grepen voor vrije transformatie moeten worden weergegeven wanneer het gereedschap actief is.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object ToolObj:

Eigenschap	Beschrijving
<code>toolObj.depth</code>	Een geheel getal dat de diepte van het gereedschap aangeeft in het pop-upmenu in het deelvenster Gereedschappen.
<code>toolObj.iconID</code>	Een geheel getal dat de bron-id van het gereedschap aangeeft.
<code>toolObj.position</code>	Alleen-lezen; een geheel getal dat de positie van het gereedschap in het deelvenster Gereedschappen aangeeft.

## toolObj.depth

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.depth
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat de diepte van het gereedschap aangeeft in het pop-upmenu in het deelvenster Gereedschappen. Deze eigenschap wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat het gereedschap diepte 1 heeft: dit betekent één niveau onder een gereedschap in het deelvenster Gereedschappen:

```
fl.tools.activeTool.depth = 1;
```

## toolObj.enablePIControl()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.enablePIControl(control, bEnable)
```

### Parameters

`control` Een tekenreeks die de naam aangeeft van het besturingselement dat moet worden in- of uitgeschakeld. Welke waarden geldig zijn, hangt af van de Eigenschapcontrole die door dit gereedschap wordt aangeroepen; zie [toolObj.setPI\(\)](#).

Eigenschapcontrole voor vormen bevat de volgende besturingselementen:

stroke	fill
--------	------

Eigenschapcontrole voor tekst bevat de volgende besturingselementen:

type	font	pointsize
color	bold	italic
direction	alignLeft	alignCenter
alignRight	alignJustify	spacing
position	autoKern	small
rotation	format	lineType
selectable	html	border
deviceFonts	varEdit	options
link	maxChars	target

Eigenschapcontrole voor films bevat de volgende besturingselementen:

size	publish	background
framerate	player	profile

**enable** Een Booleaanse waarde die bepaalt of het besturingselement moet worden ingeschakeld (`true`) of uitgeschakeld (`false`).

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt het opgegeven besturingselement in Eigenschapcontrole in- of uitgeschakeld. Wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

#### Voorbeeld

Met de volgende opdracht in het JavaScript-bestand van een uitbreidbaar gereedschap wordt Flash ingesteld om de streekoptyn in de Eigenschapcontrole voor dat gereedschap niet weer te geven:

```
theTool.enablePIControl("stroke", false);
```

## toolObj.iconID

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
toolObj.iconID
```

#### Beschrijving

Alleen-lezen eigenschap; een geheel getal met de waarde -1. Deze eigenschap wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt. De waarde -1 voor `iconID` 1 betekent dat Flash geen pictogram probeert te zoeken voor het gereedschap. In plaats daarvan moet het script voor het gereedschap aangeven welk pictogram in het deelvenster Gereedschappen moet worden weergegeven; zie [toolObj.setIcon\(\)](#).

### Voorbeeld

In het volgende voorbeeld wordt de waarde -1 (het pictogram-id van het huidige gereedschap) toegewezen aan de variabele `toolIconID`:

```
var toolIconID = fl.tools.activeTool.iconID
```

## toolObj.position

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.position
```

### Beschrijving

Alleen-lezen eigenschap; een geheel getal dat de positie van het gereedschap in het deelvenster Gereedschappen aangeeft. Deze eigenschap wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

### Voorbeeld

Met de volgende opdrachten in de methode `mouseDown()` van het JavaScript-bestand van een gereedschap wordt de positie van dat gereedschap in het deelvenster Gereedschappen weergegeven als een geheel getal in het deelvenster Uitvoer:

```
myToolPos = fl.tools.activeTool.position;  
fl.trace(myToolPos);
```

## toolObj.setIcon()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.setIcon(file)
```

### Parameters

**file** Een tekenreeks die de naam van het PNG-bestand aangeeft dat als pictogram moet worden gebruikt. Het PNG-bestand moet zich in dezelfde map bevinden als het JSFL-bestand.

### Geretoureerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een PNG-bestand geïdentificeerd dat als gereedschappictogram moet worden gebruikt in het deelvenster Gereedschappen. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat de afbeelding in het bestand PolyStar.png moet worden gebruikt als pictogram voor het gereedschap met de naam `veelhoekSter`. Deze code is afkomstig uit het voorbeeldbestand `PolyStar.jsfl` (zie [“Voorbeeld van gereedschap Veelhoek-Ster”](#) op pagina 14):

```
theTool = fl.tools.activeTool;  
theTool.setIcon("PolyStar.png");
```

## toolObj.setMenuString()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.setMenuString(menuStr)
```

### Parameters

**menuStr** Een tekenreeks die de naam aangeeft die in het pop-upmenu verschijnt als de naam voor het gereedschap.

### Geretoureerde waarde

Niets.

### Beschrijving

Methode; identificeert de tekenreeks ingesteld die in het pop-upmenu verschijnt als de naam voor het gereedschap. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

### Voorbeeld

In het volgende voorbeeld wordt aangegeven dat voor het gereedschap `theTool` de naam “PolyStar Tool” moet worden weergegeven in het pop-upmenu. Deze code is afkomstig uit het voorbeeldbestand `PolyStar.jsfl` (zie [“Voorbeeld van gereedschap Veelhoek-Ster”](#) op pagina 14):

```
theTool = fl.tools.activeTool;  
theTool.setMenuString("PolyStar Tool");
```

## toolObj.setOptionsFile()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.setOptionsFile(xmlFile)
```

### Parameters

**xmlFile** Een tekenreeks die de naam aangeeft van het XML-bestand dat de beschrijving bevat van de opties van het gereedschap. Het XML-bestand moet zich in dezelfde map bevinden als het JSFL-bestand.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een XML-bestand gekoppeld aan het gereedschap. In het bestand worden de opties opgegeven die moeten worden weergegeven in een modaal deelvenster dat wordt aangeroepen door een knop Opties in de Eigenschapcontrole. U gebruikt deze methode normaal gesproken in de functie `configureTool()` in het JSFL-bestand. Zie `configureTool()`.

Het bestand PolyStar.xml geeft bijvoorbeeld drie opties op die zijn gekoppeld aan het gereedschap Veelhoek:

```
<properties>
  <property name="Style"
    variable="style"
    list="polygon,star"
    defaultValue="0"
    type="Strings"/>

  <property name="Number of Sides"
    variable="nsides"
    min="3"
    max="32"
    defaultValue="5"
    type="Number" />

  <property name="Star point size"
    variable="pointParam"
    min="0"
    max="1"
    defaultValue=".5"
    type="Double" />
</properties>
```

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat het bestand met de naam PolyStar.xml wordt gekoppeld aan het actieve gereedschap. Deze code is afkomstig uit het voorbeeldbestand PolyStar.jsfl (zie “[Voorbeeld van gereedschap Veelhoek-Ster](#)” op pagina 14):

```
theTool = fl.tools.activeTool;
theTool.setOptionsFile("PolyStar.xml");
```

## toolObj.setPI()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.setPI(pi)
```

### Parameters

**pi** Een tekenreeks die aangeeft welke Eigenschapcontrole voor dit gereedschap moet worden aangeroepen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een specifieke Eigenschapcontrole ingesteld die moet worden gebruikt wanneer het gereedschap wordt geactiveerd. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt. Acceptabele waarden zijn "shape" (de standaardwaarde), "text" en "movie".

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat de Eigenschapcontrole voor vormen moet worden gebruikt wanneer het gereedschap wordt geactiveerd. Deze code is afkomstig uit het voorbeeldbestand PolyStar.jsfl (zie "[Voorbeeld van gereedschap Veelhoek-Ster](#)" op pagina 14):

```
theTool = fl.tools.activeTool;  
theTool.setPI("shape");
```

## toolObj.setToolName()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.setToolName(name)
```

### Parameters

**name** Een tekenreeks die de naam van het gereedschap aangeeft.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een naam aan het gereedschap toegewezen voor de configuratie van het deelvenster Gereedschappen. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt. De naam wordt alleen gebruikt door het XML-opmaakbestand dat Flash leest om het deelvenster Gereedschappen te bouwen. De naam wordt niet weergegeven in de gebruikersinterface van Flash.

### Voorbeeld

In het volgende voorbeeld wordt de naam `polystar` toegewezen aan het gereedschap met de naam `theTool`. Deze code is afkomstig uit het voorbeeldbestand PolyStar.jsfl (zie "[Voorbeeld van gereedschap Veelhoek-Ster](#)" op pagina 14):

```
theTool = fl.tools.activeTool;  
theTool.setToolName("polystar");
```



## toolObj.setToolTip()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.setToolTip(toolTip)
```

### Parameters

**toolTip** Een tekenreeks die opgeeft welke knopinfo voor het gereedschap moet worden gebruikt.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt stelt de knopinfo ingesteld die wordt weergegeven wanneer de muis boven het gereedschappictogram beweegt. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

### Voorbeeld

In het volgende voorbeeld wordt opgegeven dat de knopinfo voor het gereedschap `PolyStar Tool` moet zijn. Deze code is afkomstig uit het voorbeeldbestand `PolyStar.jsfl` (zie "[Voorbeeld van gereedschap Veelhoek-Ster](#)" op pagina 14):

```
theTool = fl.tools.activeTool;  
theTool.setToolTip("PolyStar Tool");
```

## toolObj.showPIControl()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
toolObj.showPIControl(control, bShow)
```

### Parameters

**control** Een tekenreeks die de naam aangeeft van het besturingselement dat moet worden weergegeven of verborgen. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt. Welke waarden geldig zijn, hangt af van de Eigenschapcontrole die door dit gereedschap wordt aangeroepen. Zie `toolObj.setPI()`.

Eigenschapcontrole voor vormen bevat de volgende besturingselementen:

stroke	fill
--------	------

Eigenschapcontrole voor tekst bevat de volgende besturingselementen:

type	font	pointsize
color	bold	italic
direction	alignLeft	alignCenter
alignRight	alignJustify	spacing
position	autoKern	small
rotation	format	lineType
selectable	html	border
deviceFonts	varEdit	options
link	maxChars	target

Eigenschapcontrole voor films bevat de volgende besturingselementen:

size	publish	background
framerate	player	profile

**bShow** Een Booleaanse waarde die bepaalt of het opgegeven besturingselement moet worden weergegeven (`true`) of verborgen (`false`).

#### Geretourneerde waarde

Niets.

#### Beschrijving

Methode; hiermee wordt een besturingselement in Eigenschapcontrole weergegeven of verborgen. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

#### Voorbeeld

Met de volgende opdracht in het JavaScript-bestand van een uitbreidbaar gereedschap wordt ingesteld dat de vullingsopties in de Eigenschapcontrole voor dat gereedschap niet worden weergegeven:

```
fl.tools.activeTool.showPControl("fill", false);
```

## toolObj.showTransformHandles()

#### Beschikbaarheid

Flash MX 2004.

#### Gebruik

```
toolObj.showTransformHandles(bShow)
```

#### Parameters

**bShow** Een Booleaanse waarde die bepaalt of de grepen voor vrije transformatie voor het huidige gereedschap worden weergegeven (`true`) of niet (`false`).

**Geretourneerde waarde**

Niets.

**Beschrijving**

Methode; wordt aangeroepen in de methode `configureTool()` van het JavaScript-bestand van een uitbreidbaar gereedschap om aan te geven dat de grepen voor vrije transformatie moeten worden weergegeven wanneer het gereedschap actief is. Deze methode wordt alleen gebruikt wanneer u uitbreidbare gereedschappen maakt.

**Voorbeeld**

Zie `configureTool()`.

# Hoofdstuk 47: Object Tools

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Tools kan worden benaderd vanuit het object flash (`fl.tools`). De eigenschap `tools.toolObjs` bevat een array met objecten van het type ToolObj, en de eigenschap `tools.activeTool` retourneert het object ToolObj voor het momenteel actieve gereedschap. (Zie ook [Object ToolObj](#) en de lijst met uitbreidbare gereedschappen in “[Functies en methoden op hoofdniveau](#)” op pagina 15.)

**Opmerking:** De volgende methoden en eigenschappen worden alleen gebruikt voor het maken van uitbreidbare gereedschappen.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object Tools:

Methode	Beschrijving
<code>tools.constrainPoint()</code>	Hiermee worden twee punten opgehaald en een nieuw aangepast of beperkt punt geretourneerd.
<code>tools.getKeyDown()</code>	Hiermee wordt de toets geretourneerd die het laatst is ingedrukt.
“ <code>tools.setCreatingBbox()</code> ” op pagina 522	PLACEHOLDER
<code>tools.setCursor()</code>	Hiermee wordt een bepaalde weergave ingesteld voor de muisaanwijzer.
<code>tools.snapPoint()</code>	Hiermee wordt een punt opgehaald als invoer en een nieuw punt geretourneerd dat kan worden aangepast of <i>uitgelijnd</i> op het dichtstbijzijnde geometrische object.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Tools:

Eigenschap	Beschrijving
<code>tools.activeTool</code>	Alleen-lezen; hiermee wordt het <a href="#">Object ToolObj</a> geretourneerd voor het momenteel actieve gereedschap.
<code>tools.altIsDown</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of de Alt-toets wordt ingedrukt.
<code>tools.ctrlIsDown</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of de Ctrl-toets wordt ingedrukt.
<code>tools.mouseIsDown</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of de linkermuisknop wordt ingedrukt.
<code>tools.penDownLoc</code>	Alleen-lezen; een punt dat de positie vertegenwoordigt van de laatste mousedown-gebeurtenis op het werkgebied.
<code>tools.penLoc</code>	Alleen-lezen; een tekenreeks die de huidige locatie van de muis vertegenwoordigt.
<code>tools.shiftIsDown</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of de Shift-toets wordt ingedrukt.
<code>tools.toolObjs</code>	Alleen-lezen; een array met objecten ToolObj.

## tools.activeTool

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.activeTool
```

### Beschrijving

Alleen-lezen eigenschap; hiermee wordt het [Object ToolObj](#) geretourneerd voor het momenteel actieve gereedschap.

### Voorbeeld

In het volgende voorbeeld wordt een object dat het momenteel actieve gereedschap vertegenwoordigt, opgeslagen in de variabele `theTool`:

```
var theTool = fl.tools.activeTool;
```

## tools.altIsDown

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.altIsDown
```

### Beschrijving

Alleen-lezen eigenschap; een Booleaanse waarde die aangeeft of de Alt-toets wordt ingedrukt. De waarde is `true` als de Alt-toets wordt ingedrukt, anders `false`.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of de Alt-toets wordt ingedrukt:

```
var isAltDown = fl.tools.altIsDown;
```

## tools.constrainPoint()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.constrainPoint(pt1, pt2)
```

### Parameters

`pt1`, `pt2` Punten die het punt van de eerste klik en het punt waarnaar wordt gesleept, opgeven.

### Geretourneerde waarde

Een nieuw aangepast of beperkt punt.

### Beschrijving

Methode; hiermee worden twee punten opgehaald en een nieuw aangepast of beperkt punt geretourneerd. Als de Shift-toets wordt ingedrukt op het moment dat de opdracht wordt uitgevoerd, wordt het geretourneerde punt beperkt tot het volgen van een 45°-beperking (nuttig voor bijvoorbeeld een lijn met een pijlpunt) of wordt een object beperkt zodat de hoogte-breedteverhouding wordt behouden (bijvoorbeeld voor het tekenen van een perfect vierkant met het gereedschap Rechthoek).

### Voorbeeld

In het volgende voorbeeld wordt een beperkt punt geretourneerd.

```
pt2 = fl.tools.constrainPoint(pt1, tempPt);
```

## tools.ctIsDown

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.ctIsDown
```

### Beschrijving

Alleen-lezen eigenschap; een Booleaanse waarde die `true` is als de Ctrl-toets wordt ingedrukt, anders `false`.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of de Ctrl-toets wordt ingedrukt:

```
var isCtrlDown = fl.tools.ctrlIsDown;
```

## tools.getKeyDown()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.getKeyDown()
```

### Parameters

Geen.

### Geretourneerde waarde

De gehele-getalwaarde van de toets.

### Beschrijving

Methode; hiermee wordt de toets geretourneerd die het laatst is ingedrukt.

### Voorbeeld

In het volgende voorbeeld wordt de gehele-getalwaarde van de toets geretourneerd die het laatst is ingedrukt:

```
var theKey = fl.tools.getKeyDown();  
fl.trace(theKey);
```

## tools.mouseIsDown

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.mouseIsDown
```

### Beschrijving

Alleen-lezen eigenschap; een Booleaanse waarde die `true` is als de linkermuisknop wordt ingedrukt, anders `false`.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of de linkermuisknop wordt ingedrukt:

```
var isMouseDown = fl.tools.mouseIsDown;
```

## tools.penDownLoc

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.penDownLoc
```

### Beschrijving

Alleen-lezen eigenschap; een punt dat de positie vertegenwoordigt van de laatste mousedown-gebeurtenis op het werkgebied. De eigenschap `tools.penDownLoc` omvat twee eigenschappen: `x` en `y` die corresponderen met de `x`, `y`-locatie van de muisaanwijzer.

### Voorbeeld

In het volgende voorbeeld wordt de positie van de laatste mousedown-gebeurtenis op het werkgebied bepaald en worden de `x`- en `y`-waarden weergegeven in het deelvenster Uitvoer:

```
var pt1 = fl.tools.penDownLoc;  
fl.trace("x,y location of last mouseDown event was " + pt1.x + ", " + pt1.y)
```

### Zie ook

[tools.penLoc](#)

## tools.penLoc

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`tools.penLoc`

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die de huidige locatie van de muisaanwijzer vertegenwoordigt. De eigenschap `tools.penLoc` omvat twee eigenschappen: *x* en *y* die corresponderen met de *x,y*-locatie van de muisaanwijzer.

### Voorbeeld

In het volgende voorbeeld wordt de huidige locatie van de muis bepaald:

```
var tempPt = fl.tools.penLoc;
```

### Zie ook

[tools.penDownLoc](#)

## tools.setCreatingBbox()

### Beschikbaarheid

Flash 11.

### Gebruik

`tools.setCreatingBbox()`

### Parameters

**Placeholder** Een geheel getal dat de weergave van de muisaanwijzer definieert, zoals beschreven in de volgende lijst:

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een bepaalde weergave ingesteld voor de muisaanwijzer.

### Voorbeeld

In het volgende voorbeeld wordt de muisaanwijzer ingesteld op een zwarte pijl.

```
fl.tools.setCursor(1);
```



## tools.setCursor()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.setCursor(cursor)
```

### Parameters

**cursor** Een geheel getal dat de weergave van de muisaanwijzer definieert, zoals beschreven in de volgende lijst:

- 0 = plus-cursor (+)
- 1 = zwarte pijl
- 2 = witte pijl
- 3 = pijl met vier richtingen
- 4 = horizontale pijl met twee richtingen
- 5 = verticale pijl met twee richtingen
- 6 = X
- 7 = hand-cursor

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt een bepaalde weergave ingesteld voor de muisaanwijzer.

### Voorbeeld

In het volgende voorbeeld wordt de muisaanwijzer ingesteld op een zwarte pijl.

```
fl.tools.setCursor(1);
```

## tools.shiftIsDown

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.shiftIsDown
```

### Beschrijving

Alleen-lezen eigenschap; een Booleaanse waarde die `true` is als de Shift-toets wordt ingedrukt, anders `false`.

### Voorbeeld

In het volgende voorbeeld wordt bepaald of de Shift-toets wordt ingedrukt.

```
var isShiftDown = fl.tools.shiftIsDown;
```

## tools.snapPoint()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.snapPoint(pt)
```

### Parameters

**pt** Hiermee wordt de locatie opgegeven van het punt waarvoor u een uitlijnpunt wilt retourneren.

### Geretourneerde waarde

Een nieuw punt dat kan worden aangepast of uitgelijnd op het dichtstbijzijnde geometrische object.

### Beschrijving

Methode; hiermee wordt een punt opgehaald als invoer en een nieuw punt geretourneerd dat kan worden aangepast of *uitgelijnd* op het dichtstbijzijnde geometrische object. Als uitlijnen is uitgeschakeld in het menu Beeld van de Flash gebruikersinterface, wordt het oorspronkelijke punt geretourneerd.

### Voorbeeld

In het volgende voorbeeld wordt een nieuw punt geretourneerd dat kan worden uitgelijnd op het dichtstbijzijnde geometrische object.

```
var theSnapPoint = fl.tools.snapPoint(pt1);
```

## tools.toolObjs

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
tools.toolObjs
```

### Beschrijving

Alleen-lezen eigenschap; een array met objecten van het type ToolObj (zie [Object ToolObj](#)).

# Hoofdstuk 48: Object Vertex

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object Vertex is het deel van de gegevensstructuur van een vorm dat de coördinaatgegevens bevat.

## Overzicht van methoden

U kunt de volgende methoden gebruiken met het object Vertex:

Methoden	Beschrijving
<code>vertex.getHalfEdge()</code>	Hiermee wordt een <a href="#">Object HalfEdge</a> opgehaald dat dit hoekpunt deelt.
<code>vertex.setLocation()</code>	Hiermee wordt de locatie ingesteld van het hoekpunt.

## Overzicht van eigenschappen

De volgende eigenschappen zijn beschikbaar voor het object Vertex:

Eigenschap	Beschrijving
<code>vertex.x</code>	Alleen-lezen; de x-locatie van het hoekpunt in pixels.
<code>vertex.y</code>	Alleen-lezen; de y-locatie van het hoekpunt in pixels.

## vertex.getHalfEdge()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`vertex.getHalfEdge()`

### Parameters

Geen.

### Geretourneerde waarde

Een [Object HalfEdge](#).

### Beschrijving

Methoden; hiermee wordt een [Object HalfEdge](#) opgehaald dat dit hoekpunt deelt.

### Voorbeeld

In het volgende voorbeeld wordt getoond hoe u andere halve randen ophaalt die hetzelfde hoekpunt delen:

**Object Vertex**

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var theVertex = hEdge.getVertex();
var someHEdge = theVertex.getHalfEdge(); // Not necessarily the same half edge
var theSameVertex = someHEdge.getVertex();
fl.trace('the same vertex: ' + theSameVertex);
```

## vertex.setLocation()

**Beschikbaarheid**

Flash MX 2004.

**Gebruik**

```
vertex.setLocation(x, y)
```

**Parameters**

**x** Een drijvende-kommawaarde die de *x*-coördinaat opgeeft waar het hoekpunt moet worden gepositioneerd, in pixels.

**y** Een drijvende-kommawaarde die de *y*-coördinaat opgeeft waar het hoekpunt moet worden gepositioneerd, in pixels.

**Geretourneerde waarde**

Niets.

**Beschrijving**

Methode; hiermee wordt de locatie van het hoekpunt ingesteld. U moet `shape.beginEdit()` aanroepen voordat u deze methode gebruikt.

**Voorbeeld**

In het volgende voorbeeld wordt het hoekpunt ingesteld op het oorsprongpunt:

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
var hEdge = shape.edges[0].getHalfEdge(0);
var vertex = hEdge.getVertex();
var someHEdge = vertex.getHalfEdge();
var vertex = someHEdge.getVertex();
// Move the vertex to the origin.
vertex.setLocation(0.0, 0.0);
shape.endEdit();
```

## vertex.x

**Beschikbaarheid**

Flash MX 2004.

### **Gebruik**

vertex.x

### **Beschrijving**

Alleen-lezen eigenschap; de *x*-locatie van het hoekpunt, in pixels.

### **Voorbeeld**

In het volgende voorbeeld wordt de locatie van de *x*- en *y*-waarde van het hoekpunt weergegeven in het deelvenster Uitvoer:

```
var shape = fl.getDocumentDOM().selection[0];  
var hEdge = shape.edges[0].getHalfEdge(0);  
var vertex = hEdge.getVertex();  
  
fl.trace('x location of vertex is: ' + vertex.x);  
fl.trace('y location of vertex is: ' + vertex.y);
```

## **vertex.y**

### **Beschikbaarheid**

Flash MX 2004.

### **Gebruik**

vertex.y

### **Beschrijving**

Alleen-lezen eigenschap; de *y*-locatie van het hoekpunt, in pixels.

### **Voorbeeld**

Zie [vertex.x](#).

# Hoofdstuk 49: Object VideoItem

**Overerving** [Object Item](#) > object VideoItem

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Het object VideoItem is een subklasse van het [Object Item](#).

## Overzicht van methoden

Het object VideoItem heeft naast de methoden voor het object Item ook de volgende methode:

Eigenschap	Beschrijving
<code>videoItem.exportToFLV()</code>	Hiermee wordt het opgegeven item geëxporteerd naar een FLV-bestand.

## Overzicht van eigenschappen

Naast de eigenschappen van het object Item kunt u de volgende eigenschappen gebruiken voor het object VideoItem:

Eigenschap	Beschrijving
<code>videoItem.fileLastModifiedDate</code>	Alleen-lezen; een tekenreeks die een hexadecimaal getal bevat dat het aantal seconden vertegenwoordigt dat is verstreken tussen 1 januari 1970 en de wijzigingsdatum van het oorspronkelijke bestand (op schijf) op het moment dat het bestand is geïmporteerd naar de bibliotheek.
<code>videoItem.sourceFileExists</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of het bestand dat naar de bibliotheek is geïmporteerd, nog bestaat op de locatie vanwaar het werd geïmporteerd.
<code>videoItem.sourceFileIsCurrent</code>	Alleen-lezen; een Booleaanse waarde die aangeeft of de wijzigingsdatum van het bibliotheekitem gelijk is aan de wijzigingsdatum (op schijf) van het bestand dat is geïmporteerd.
<code>videoItem.sourceFilePath</code>	Alleen-lezen; een tekenreeks die het pad naar het video-item aangeeft.
<code>videoItem.videoType</code>	Alleen-lezen; een tekenreeks die het type video aangeeft.

## videoItem.exportToFLV()

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
videoItem.exportToFLV(fileURI)
```

### Parameters

`fileURI` Een tekenreeks, uitgedrukt als bestand:/// URI, die het pad en de naam van het geëxporteerde bestand aangeeft.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het bestand is geëxporteerd, anders `false`.

### Beschrijving

Methode; hiermee wordt het opgegeven item geëxporteerd naar een FLV-bestand.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een video-item is, wordt dit item met de volgende code geëxporteerd als een FLV-bestand:

```
var videoFileURL = "file:///C:/out.flv";  
var libItem = fl.getDocumentDOM().library.items[0];  
libItem.exportToFLV(videoFileURL);
```

## videoItem.fileLastModifiedDate

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
videoItem.fileLastModifiedDate
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks die een hexadecimaal getal bevat dat het aantal seconden vertegenwoordigt dat is verstreken tussen 1 januari 1970 en de wijzigingsdatum van het oorspronkelijke bestand (op schijf) op het moment dat het bestand werd geïmporteerd naar de bibliotheek. Als het bestand niet meer bestaat, is deze waarde "00000000".

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een video-item is, wordt met de volgende code een hexadecimaal getal weergegeven zoals hierboven beschreven.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

### Zie ook

[videoItem.sourceFileExists](#), [videoItem.sourceFileIsCurrent](#), [videoItem.sourceFilePath](#), [FLfile.getModificationDate\(\)](#)

## videoItem.sourceFileExists

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
videoItem.sourceFileExists
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde: `true` als het bestand dat werd geïmporteerd in de bibliotheek, nog bestaat op de locatie vanwaar het werd geïmporteerd, anders `false`.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een video-item is, wordt met de volgende code "true" weergegeven als het bestand dat naar de bibliotheek werd geïmporteerd, nog bestaat.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

### Zie ook

[videoItem.sourceFileIsCurrent](#), [videoItem.sourceFilePath](#)

## videoItem.sourceFileIsCurrent

### Beschikbaarheid

Flash CS4 Professional.

### Gebruik

```
videoItem.sourceFileIsCurrent
```

### Beschrijving

Alleen-lezen eigenschap: een Booleaanse waarde: `true` als de wijzigingsdatum van het bibliotheekitem gelijk is aan de wijzigingsdatum (op schijf) van het bestand dat werd geïmporteerd, anders `false`.

### Voorbeeld

Ervan uitgaande dat het eerste item in de bibliotheek een video-item is, wordt met de volgende code "true" weergegeven als het bestand dat werd geïmporteerd, niet op schijf is gewijzigd sinds het werd geïmporteerd.

```
var libItem = fl.getDocumentDOM().library.items[0];  
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

### Zie ook

[videoItem.fileLastModifiedDate](#), [videoItem.sourceFilePath](#)

## videoItem.sourceFilePath

### Beschikbaarheid

Flash 8.

### Gebruik

```
videoItem.sourceFilePath
```

### Beschrijving

Alleen-lezen eigenschap: een tekenreeks, uitgedrukt als een bestand:/// URI die het pad naar het video-item aangeeft.



### Voorbeeld

In het onderstaande voorbeeld worden de naam en het pad naar het bronbestand weergegeven van items in de bibliotheek van het type `video`:

```
for (idx in fl.getDocumentDOM().library.items) {  
  if (fl.getDocumentDOM().library.items[idx].itemType == "video") {  
    var myItem = fl.getDocumentDOM().library.items[idx];  
    fl.trace(myItem.name + " source is " + myItem.sourceFilePath);  
  }  
}
```

### Zie ook

[videoItem.sourceFileExists](#)

## videoItem.videoType

### Beschikbaarheid

Flash 8.

### Gebruik

`videoItem.videoType`

### Beschrijving

Alleen-lezen eigenschap; een tekenreeks die het type video aangeeft. Mogelijke waarden zijn "embedded video", "linked video" en "video".

### Voorbeeld

In het onderstaande voorbeeld worden de naam en het type weergegeven van items in de bibliotheek van het type `video`:

```
for (idx in fl.getDocumentDOM().library.items) {  
  if (fl.getDocumentDOM().library.items[idx].itemType == "video") {  
    var myItem = fl.getDocumentDOM().library.items[idx];  
    fl.trace(myItem.name + " is " + myItem.videoType);  
  }  
}
```

# Hoofdstuk 50: Object XMLUI

## Beschikbaarheid

Flash MX 2004.

## Beschrijving

Flash 8 ondersteunt aangepaste dialoogvensters die zijn geschreven in een subset van de XML User Interface Language (XUL). Een XMLUI-dialoogvenster (XML User Interface) kan door verschillende functies van Flash worden gebruikt, zoals opdrachten en gedragingen, om u een gebruikersinterface te bieden voor functies die u met uitbreidbaarheid maakt. Met het object XMLUI kunt u eigenschappen van een XMLUI-dialoogvenster ophalen en instellen en een XMLUI-dialoogvenster accepteren of annuleren. De XMLUI-methoden kunnen worden gebruikt in callbacks, zoals `oncommand`-handlers in knoppen.

U kunt een bestand `dialog.xml` schrijven en dit vanuit de JavaScript-API aanroepen met de methode `document.xmlPanel()`. Als u object wilt ophalen dat het huidige XMLUI-dialoogvenster vertegenwoordigt, gebruikt u `fl.xmlui`.

## Overzicht van methoden

De volgende methoden zijn beschikbaar voor het object XMLUI:

Methoden	Beschrijving
<code>xmlui.accept()</code>	Hiermee wordt het huidige dialoogvenster XMLUI gesloten met de status "accept".
<code>xmlui.cancel()</code>	Hiermee wordt het huidige dialoogvenster XMLUI gesloten met de status "cancel".
<code>xmlui.get()</code>	Hiermee wordt de waarde van de opgegeven eigenschap van het huidige dialoogvenster XMLUI opgehaald.
<code>xmlui.getControlItemElement()</code>	Hiermee wordt het huidige besturingselementitem geretourneerd voor het opgegeven besturingselement.
<code>xmlui.setEnabled()</code>	Hiermee wordt een Booleaanse waarde geretourneerd die aangeeft of het besturingselement is ingeschakeld of uitgeschakeld (grijs weergegeven).
<code>xmlui.setVisible()</code>	Hiermee wordt een Booleaanse waarde geretourneerd die aangeeft of het besturingselement zichtbaar of verborgen is.
<code>xmlui.set()</code>	Hiermee wordt de waarde van de opgegeven eigenschap van het huidige dialoogvenster XMLUI gewijzigd.
<code>xmlui.setControlItemElement()</code>	Hiermee worden het label en de waarde voor het huidige item ingesteld.
<code>xmlui.setControlItemElements()</code>	Hiermee worden het label en de waardeparen voor het huidige item ingesteld.
<code>xmlui.setEnabled()</code>	Hiermee wordt een besturingselement in- of uitgeschakeld (grijs weergegeven).
<code>xmlui.setVisible()</code>	Hiermee wordt het besturingselement weergegeven of verborgen.

## xmlui.accept()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`xmlui.accept()`

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het huidige dialoogvenster XMLUI gesloten met de status "accept". Dit is gelijk aan een klik op de knop OK door de gebruiker.

### Zie ook

`fl.xmlui`, `document.xmlPanel()`, `xmlui.cancel()`

## xmlui.cancel()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

`xmlui.cancel()`

### Parameters

Geen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het huidige dialoogvenster XMLUI gesloten met de status "cancel". Dit is gelijk aan een klik op de knop Annuleren door de gebruiker.

### Zie ook

`fl.xmlui`, `document.xmlPanel()`, `xmlui.accept()`

## xmlui.get()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
xmlui.get(controlPropertyName)
```

### Parameters

**controlPropertyName** Een tekenreeks die de naam van de eigenschap XMLUI aangeeft waarvan u de waarde wilt ophalen.

### Geretoureerde waarde

Een tekenreeks die de waarde van de opgegeven eigenschap vertegenwoordigt. In gevallen waarin u de Booleaanse waarde `true` of `false` verwacht, wordt de tekenreeks `"true"` of `"false"` geretourneerd.

### Beschrijving

Methode; hiermee wordt de waarde van de opgegeven eigenschap van het huidige dialoogvenster XMLUI opgehaald.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van een eigenschap met de naam `URL` geretourneerd:

```
fl.xmlui.get("URL");
```

### Zie ook

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.getControlItemElement\(\)](#), [xmlui.set\(\)](#)

## xmlui.getControlItemElement()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.getControlItemElement(controlPropertyName)
```

### Parameters

**controlPropertyName** Een tekenreeks die de eigenschap aangeeft waarvan u het element besturingsselementitem wilt ophalen.

### Geretoureerde waarde

Een object dat het huidige besturingsselementitem vertegenwoordigt voor het besturingselement dat is opgegeven door *controlPropertyName*.

### Beschrijving

Methode; hiermee worden het label en de waarde geretourneerd van de geselecteerde regel in een besturingselement `ListBox` of `ComboBox` voor het besturingselement dat is opgegeven door *controlPropertyName*.

### Voorbeeld

In het volgende voorbeeld worden het label en de waarde geretourneerd van de geselecteerde regel voor het besturingselement `myListBox`:

```
var elem = new Object();  
elem = fl.xmlui.getControlItemElement("myListBox");  
fl.trace("label = " + elem.label + " value = " + elem.value);
```

### Zie ook

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.get\(\)](#), [xmlui.setControlItemElement\(\)](#),  
[xmlui.setControlItemElements\(\)](#)

## xmlui.setEnabled()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.setEnabled(controlID)
```

### Parameters

`controlID` Een tekenreeks die het kenmerk `Id` opgeeft van het besturingselement waarvan u de status wilt ophalen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` is als het besturingselement is ingeschakeld, anders `false`.

### Beschrijving

Methode; hiermee wordt een Booleaanse waarde geretourneerd die aangeeft of het besturingselement is ingeschakeld of uitgeschakeld (grijs weergegeven).

### Voorbeeld

In het volgende voorbeeld wordt een waarde geretourneerd die aangeeft of het besturingselement met het `Id`-kenmerk `myListBox` is ingeschakeld:

```
var isEnabled = fl.xmlui.setEnabled("myListBox");  
fl.trace(isEnabled);
```

### Zie ook

[fl.xmlui](#), [document.xmlPanel\(\)](#), [xmlui.setEnabled\(\)](#)

## xmlui.getVisible()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.getVisible(controlID)
```

### Parameters

**controlID** Een tekenreeks die het kenmerk Id opgeeft van het besturingselement waarvan u de zichtbaarheid wilt ophalen.

### Geretourneerde waarde

Een Booleaanse waarde: `true` als het besturingselement zichtbaar is, `false` als het onzichtbaar is (verborgen).

### Beschrijving

Methode; hiermee wordt een Booleaanse waarde geretourneerd die aangeeft of het besturingselement zichtbaar of verborgen is.

### Voorbeeld

In het volgende voorbeeld wordt een waarde geretourneerd die aangeeft of het besturingselement met het Id-kenmerk `myListBox` zichtbaar is:

```
var isVisible = fl.xmlui.getVisible("myListBox");  
fl.trace(isVisible);
```

### Zie ook

[xmlui.setVisible\(\)](#)

## xmlui.set()

### Beschikbaarheid

Flash MX 2004.

### Gebruik

```
xmlui.set(controlPropertyName, value)
```

### Parameters

**controlPropertyName** Een tekenreeks die de naam opgeeft van de eigenschap XMLUI die moet worden gewijzigd.

**value** Een tekenreeks die de waarde opgeeft waarop u de eigenschap XMLUI wilt instellen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt de waarde van de opgegeven eigenschap van het huidige dialoogvenster XMLUI gewijzigd.

### Voorbeeld

In het volgende voorbeeld wordt de waarde van een eigenschap met de naam URL ingesteld op `www.adobe.com`:

```
fl.xmlui.set("URL", "www.adobe.com");
```

### Zie ook

`fl.xmlui`, `document.xmlPanel()`, `xmlui.get()`, `xmlui.setControlItemElement()`,  
`xmlui.setControlItemElements()`

## xmlui.setControlItemElement()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.setControlItemElement(controlPropertyName, elementItem)
```

### Parameters

**controlPropertyName** Een tekenreeks die het besturingselementitem aangeeft dat moet worden ingesteld.

**elementItem** Een JavaScript-object met een tekenreeks-eigenschap met de naam `label` en een optionele tekenreeks-eigenschap met de naam `value`. Als de eigenschap `value` niet bestaat, wordt deze gemaakt en wordt er dezelfde waarde aan toegewezen als aan `label`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden het label en de waarde ingesteld van de geselecteerde regel in het besturingselement `ListBox` of `ComboBox` dat is opgegeven door *controlPropertyName*.

### Voorbeeld

In het volgende voorbeeld worden het label en de waarde voor het huidige item van de besturingselementeigenschap met de naam `PhoneNumber` ingesteld:

```
var elem = new Object();  
elem.label = "Fax";  
elem.value = "707-555-5555";  
fl.xmlui.setControlItemElement("PhoneNumber", elem);
```

### Zie ook

`fl.xmlui`, `document.xmlPanel()`, `xmlui.getControlItemElement()`, `xmlui.set()`,  
`xmlui.setControlItemElements()`

## xmlui.setControlItemElements()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.setControlItemElements(controlID, elementItemArray)
```

### Parameters

**controlID** Een tekenreeks die het kenmerk Id opgeeft van het besturingselement dat u wilt instellen.

**elementItemArray** Een array van JavaScript-objecten, waarin elk object een tekenreekseigenschap heeft met de naam `label` en een optionele tekenreekseigenschap met de naam `value`. Als de eigenschap `value` niet bestaat, wordt deze gemaakt en wordt er dezelfde waarde aan toegewezen als aan `label`.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee worden de waarden gewist van het besturingselement `ListBox` of `ComboBox` dat is opgegeven door `controlID`, en worden de lijst- of menu-items vervangen door de `label-value`-paren die zijn opgegeven door `elementItemArray`.

### Voorbeeld

In het volgende voorbeeld worden het label en de waarde van items in het besturingselement met het Id-kenmerk `myControlID` ingesteld op de `label-value`-paren die zijn opgegeven:

```
var nameArray = new Array("January", "February", "March");
var monthArray = new Array();
for (i=0;i<nameArray.length;i++){
    elem = new Object();
    elem.label = nameArray[i];
    elem.value = i;
    monthArray[i] = elem;
}
fl.xmlui.setControlItemElements("myControlID", monthArray);
```

### Zie ook

[xmlui.getControlItemElement\(\)](#), [xmlui.set\(\)](#), [xmlui.setControlItemElement\(\)](#)

## xmlui.setEnabled()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.setEnabled(controlID, enable)
```

### Parameters

**controlID** Een tekenreeks die het kenmerk Id opgeeft van het besturingselement dat u wilt in- of uitschakelen.

**enable** Een Booleaanse waarde: `true` als u het besturingselement wilt inschakelen, `false` als u het wilt uitschakelen (grijs weergeven).

### Geretourneerde waarde

Niets.



### Beschrijving

Methode; hiermee wordt een besturingselement in- of uitgeschakeld (grijs weergegeven).

### Voorbeeld

In het volgende voorbeeld wordt het besturingselement met het Id-kenmerk `myControl` grijs weergegeven:

```
fl.xmlui.setEnabled("myControl", false);
```

### Zie ook

[xmlui.setEnabled\(\)](#)

## xmlui.setVisible()

### Beschikbaarheid

Flash 8.

### Gebruik

```
xmlui.setVisible(controlID, visible)
```

### Parameters

**controlID** Een tekenreeks die het kenmerk Id opgeeft van het besturingselement dat u wilt weergeven of verbergen.

**visible** Een Booleaanse waarde: `true` als u het besturingselement wilt weergeven, `false` als u het wilt verbergen.

### Geretourneerde waarde

Niets.

### Beschrijving

Methode; hiermee wordt het besturingselement weergegeven of verborgen.

### Voorbeeld

In het volgende voorbeeld wordt het besturingselement met het Id-kenmerk `myControl` verborgen:

```
fl.xmlui.setVisible("myControl", false);
```

### Zie ook

[xmlui.setVisible\(\)](#)

# Hoofdstuk 51: Uitbreidbaarheid in C

In dit hoofdstuk wordt het mechanisme voor uitbreidbaarheid in C beschreven. Hiermee kunt u uitbreidingsbestanden van Adobe Flash CS4 Professional implementeren met een combinatie van JavaScript en eigen C-code. In deze versie van Flash zijn geen wijzigingen in het mechanisme geïntroduceerd.

## Over uitbreidbaarheid

Als u uitbreidbaarheid wilt implementeren, definieert u functies met C, bundelt ze in een Dynamic Linked Library (DLL) of een gedeelde bibliotheek, slaat de bibliotheek op in de juiste map en roept u vervolgens de functies uit JavaScript aan met de Adobe Flash JavaScript API.

U kunt bijvoorbeeld een functie definiëren waarmee krachtige berekeningen efficiënter worden uitgevoerd dan met JavaScript, waardoor de prestaties verbeteren, of u kunt meer geavanceerde hulpprogramma's of effecten maken.

Dit mechanisme voor uitbreidbaarheid is een subset van de Adobe Dreamweaver CS3 API. Als u bekend bent met deze API, herkent u mogelijk de functies in deze API voor het mechanisme voor uitbreidbaarheid in C. Deze API verschilt echter op de volgende punten van de Dreamweaver API:

- Deze API bevat niet alle opdrachten uit de Dreamweaver API.
- Alle declaraties van het type `wchar_t` en `char` in de Dreamweaver API worden geïmplementeerd als `unsigned short` declaraties in deze API, ter ondersteuning van Unicode bij het doorgeven van tekenreeksen.
- De functie `JSVal JS_BytesToValue()` in deze API is geen onderdeel van de Dreamweaver API.
- De locatie voor de opslag van het DLL-bestand of de gedeelde bibliotheekbestanden verschilt (zie “[C-functies integreren](#)” op pagina 540).

## C-functies integreren

Met het mechanisme voor uitbreidbaarheid in C kunt u Flash-uitbreidingsbestanden implementeren met een combinatie van JavaScript en C-code. Het proces voor de implementatie van deze voorziening wordt samengevat in de volgende stappen:

- 1 Definieer functies met de taal C of C++.
- 2 Bundel de functies in een DLL-bestand (Windows) of een gedeelde bibliotheek (Macintosh).
- 3 Sla het DLL-bestand of de bibliotheek op in de juiste map:
  - Windows 7:  
`opstartstation\Users\gebruikersnaam\AppData\Adobe\FIash CS5 of CS5.5\taal\Configuration\External Libraries`
  - Windows Vista:  
`opstartstation\Users\gebruikersnaam\Local Settings\Application Data\Adobe\FIash CS5 of CS5.5\taal\Configuration\External Libraries`
  - Windows XP:

*opstartstation*\Documents and Settings\*gebruikersnaam*\Local Settings\Application Data\Adobe\Flash CS5 of CS5.5\*taal*\Configuration\External Libraries

- Mac OS X:

Macintosh HD/Users/*gebruikersnaam*/Library/Application Support/Adobe/Flash CS5 of CS5.5/*taal*/Configuration/External Libraries

4 Maak een JSFL-bestand waarmee de functies worden aangeroepen.

5 Voer het JSFL-bestand uit via het menu Opdrachten in de Flash-ontwerpomgeving.

Zie “[Voorbeeld-DLL-implementatie](#)” op pagina 544 voor meer informatie.

## Uitbreidbaarheid in C en de JavaScript-interpretter

De C-code in het DLL-bestand of de gedeelde bibliotheek communiceert op drie verschillende momenten met de Flash JavaScript API:

- Tijdens het opstarten om de functies van de bibliotheek te registreren
- Tijdens het aanroepen van de C-functie om de argumenten uit te pakken die door JavaScript worden doorgegeven aan C
- Voordat de C-functie een waarde retourneert om deze waarde aan een pakket toe te voegen

Voor het uitvoeren van deze taken definieert de interpreter enkele gegevenstypen en wordt er een API weergegeven. Definities voor de gegevenstypen en functies in deze sectie worden weergegeven in het bestand `mm_jsapi.h`. De bibliotheek werkt alleen naar behoren als u het bestand `mm_jsapi.h` toevoegt boven aan elk bestand in de bibliotheek. Gebruik hiervoor de volgende regel:

```
#include "mm_jsapi.h"
```

Als u het bestand `mm_jsapi.h` toevoegt, wordt ook het bestand `mm_jsapi_environment.h` toegevoegd, waarmee de structuur `MM_Environment` wordt gedefinieerd.

U kunt het bestand `mm_jsapi.h` extraheren uit het voorbeeld-ZIP- of SIT-bestand (zie “[Voorbeeld-DLL-implementatie](#)” op pagina 544) of de volgende code kopiëren naar een bestand dat u de naam `mm_jsapi.h` geeft:

```
#ifndef _MM_JSAPI_H_
#define _MM_JSAPI_H_

/*****
 * Public data types
 *****/

typedef struct JSContext JSContext;
typedef struct JSObject JSObject;
typedef long jsval;
#ifndef JSBool
typedef long JSBool;
#endif

typedef JSBool (*JSNative)(JSContext *cx, JSObject *obj, unsigned int argc,
jsval *argv, jsval *rval);

/* Possible values for JSBool */
#define JS_TRUE 1
#define JS_FALSE 0
```

```

/*****
 * Public functions
 *****/

/* JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs) */
#define JS_DefineFunction(n, c, a) \
(mmEnv.defineFunction ? (*(mmEnv.defineFunction))(mmEnv.libObj, n, c, a) \
: JS_FALSE)

/* unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength) */
#define JS_ValueToString(c, v, l) \
(mmEnv.valueToString? (*(mmEnv.valueToString))(c, v, l) : (char *)0)

/* unsigned char *JS_ValueToBytes(JSContext *cx, jsval v, unsigned int *pLength) */
#define JS_ValueToBytes(c, v, l) \
(mmEnv.valueToBytes? (*(mmEnv.valueToBytes))(c, v, l) : (unsigned char *)0)

/* JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp); */
#define JS_ValueToInteger(c, v, l) \
(mmEnv.valueToInteger ? (*(mmEnv.valueToInteger))(c, v, l) : JS_FALSE)

/* JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp); */
#define JS_ValueToDouble(c, v, d) \
(mmEnv.valueToDouble? (*(mmEnv.valueToDouble))(c, v, d) : JS_FALSE)

/* JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp); */
#define JS_ValueToBoolean(c, v, b) \
(mmEnv.valueToBoolean ? (*(mmEnv.valueToBoolean))(c, v, b) : JS_FALSE)

/* JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op); */
#define JS_ValueToObject(c, v, o) \
(mmEnv.valueToObject? (*(mmEnv.valueToObject))(c, v, o) : JS_FALSE)

/* JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp); */
#define JS_StringToValue(c, b, s, v) \
(mmEnv.stringToValue? (*(mmEnv.stringToValue))(c, b, s, v) : JS_FALSE)

/* JSBool JS_BytesToValue(JSContext *cx, unsigned char *bytes, uint sz, jsval *vp); */
#define JS_BytesToValue(c, b, s, v) \
(mmEnv.bytesToValue? (*(mmEnv.bytesToValue))(c, b, s, v) : JS_FALSE)

/* JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp); */
#define JS_DoubleToValue(c, d, v) \
(mmEnv.doubleToValue? (*(mmEnv.doubleToValue))(c, d, v) : JS_FALSE)

/* jsval JS_IntegerToValue(long lv); */
#define JS_IntegerToValue(lv) (((jsval)(lv) << 1) | 0x1)

/* jsval JS_BooleanToValue(JSBool bv); */
#define JS_BooleanToValue(bv) (((jsval)(bv) << 3) | 0x6)

/* jsval JS_ObjectToValue(JSObject *obj); */
#define JS_ObjectToValue(ov) ((jsval)(ov))

/* unsigned short *JS_ObjectType(JSObject *obj); */

```

```

#define JS_ObjectType(o) \
(mmEnv.objectType ? *(mmEnv.objectType))(o) : (char *)0

/* JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length, jsval *v) */
#define JS_NewArrayObject(c, l, v) \
(mmEnv.newArrayObject ? *(mmEnv.newArrayObject))(c, l, v) : (JSObject *)0

/* long JS_GetArrayLength(JSContext *cx, JSObject *obj) */
#define JS_GetArrayLength(c, o) \
(mmEnv.getArrayLength ? *(mmEnv.getArrayLength))(c, o) : -1

/* JSBool JS_GetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp) */
#define JS_GetElement(c, o, i, v) \
(mmEnv.getElement ? *(mmEnv.getElement))(c, o, i, v) : JS_FALSE

/* JSBool JS_SetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp) */
#define JS_SetElement(c, o, i, v) \
(mmEnv.setElement ? *(mmEnv.setElement))(c, o, i, v) : JS_FALSE

/* JSBool JS_ExecuteScript(JSContext *cx, JSObject *obj, unsigned short *script,
 * unsigned int sz, jsval *rval) */
#define JS_ExecuteScript(c, o, s, z, r) \
(mmEnv.executeScript? *(mmEnv.executeScript))(c, o, s, z, (LPCTSTR)___FILE__, \
___LINE__, r) : JS_FALSE

/* JSBool JS_ReportError(JSContext *cx, unsigned short *error, unsigned int sz) */
#define JS_ReportError(c, e, s) \
(mmEnv.reportError? *(mmEnv.reportError))(c, e, s) : JS_FALSE

/*****
 * Private data types, macros, and globals
 *****/

typedef struct {
JSObject *libObj;
JSBool (*defineFunction)(JSObject *libObj, unsigned short *name, JSNative call,
unsigned int nargs);
unsigned short (*valueToString)(JSContext *cx, jsval v, unsigned int *pLength);
unsigned char *(*valueToBytes)(JSContext *cx, jsval v, unsigned int *pLength);
JSBool (*valueToInteger)(JSContext *cx, jsval v, long *lp);
JSBool (*valueToDouble)(JSContext *cx, jsval v, double *dp);
JSBool (*valueToBoolean)(JSContext *cx, jsval v, JSBool *bp);
JSBool (*valueToObject)(JSContext *cx, jsval v, JSObject **op);
JSBool (*stringToValue)(JSContext *cx, unsigned short *b, unsigned int sz, jsval *vp);
JSBool (*bytesToValue)(JSContext *cx, unsigned char *b, unsigned int sz, jsval *vp);
JSBool (*doubleToValue)(JSContext *cx, double dv, jsval *vp);
unsigned short *(*objectType)(JSObject *obj);
JSObject *(*newArrayObject)(JSContext *cx, unsigned int length, jsval *vp);
long (*getArrayLength)(JSContext *cx, JSObject *obj);
JSBool (*getElement)(JSContext *cx, JSObject *obj, unsigned int idx,
jsval *vp);
JSBool (*setElement)(JSContext *cx, JSObject *obj, unsigned int idx,
jsval *vp);
JSBool (*executeScript)(JSContext *cx, JSObject *obj, unsigned short *script,
unsigned int sz, unsigned short *file, unsigned int lineNum, jsval *rval);

```

```

JSBool (*reportError)(JSContext *cx, unsigned short *error, unsigned int sz);
} MM_Environment;

extern MM_Environment mmEnv;

// Declare the external entry point and linkage
#ifdef _WIN32
# ifndef _MAC
// Windows
__declspec( dllexport ) void MM_InitWrapper( MM_Environment *env, unsigned int envSize );
# endif
#else
extern void MM_InitWrapper( MM_Environment *env, unsigned int envSize );
#endif

#define MM_STATE\
/* Definitions of global variables */ \
MM_Environment mmEnv; \
\
void\
MM_InitWrapper(MM_Environment *env, unsigned int envSize) \
{ \
extern void MM_Init();\
\
char **envPtr = (char **)env; \
char **mmPtr = (char **)(&mmEnv);\
char **envEnd = (char **)((char *)envPtr + envSize);\
char **mmEnd = (char **)((char *)mmPtr+ sizeof(MM_Environment)); \
\
/* Copy fields from env to mmEnv, one pointer at a time */\
while (mmPtr < mmEnd && envPtr < envEnd)\
*mmPtr++ = *envPtr++; \
\
/* If env doesn't define all of mmEnv's fields, set extras to NULL */ \
while (mmPtr < mmEnd) \
*mmPtr++ = (char *)0; \
\
/* Call user's MM_Init function */\
MM_Init();\
} \

#endif /* _MM_JSAPI_H_ */

```

## Voorbeeld-DLL-implementatie

In deze sectie wordt beschreven hoe u een eenvoudige DLL-implementatie uitvoert. Als u wilt zien hoe het proces werkt zonder daadwerkelijk het DLL-bestand te maken, installeert u de voorbeeld-DLL-bestanden uit het bestand Samples.zip (de bestanden staan in de map ExtendingFlash/dllSampleComputeSum). (Zie “[Voorbeeldimplementaties](#)” op pagina 13 voor informatie over het downloaden van het bestand Samples.zip.) Extraheer de voorbeeldbestanden uit het bestand dllSampleComputeSum.dmg of dllSampleComputeSum.zip en voer de volgende handelingen uit:

- Sla het bestand Sample.jsfl op in de map Configuration/Commands (zie “[JSFL-bestanden opslaan](#)” op pagina 2).

- Sla het bestand Sample.dll op in de map Configuration/External Libraries (zie “[C-functies integreren](#)” op pagina 540).
- Selecteer in de Flash-ontwerpomgeving Opdrachten > Voorbeeld. De traceringsinstructie in het JSFL-bestand stuurt de resultaten van de functie die in Sample.dll is gedefinieerd naar het uitvoervenster.

In de rest van deze sectie wordt de ontwikkeling van het voorbeeld besproken. In dit geval bevat het DLL-bestand slechts één functie waarmee twee getallen worden opgeteld. De C-code wordt weergegeven in het volgende voorbeeld:

```
// Source code in C
// Save the DLL or shared library with the name "Sample".
#include <windows.h>
#include <stdlib.h>

#include "mm_jsapi.h"

// A sample function
// Every implementation of a JavaScript function must have this signature.
JSBool computeSum(JSContext *cx, JSObject *obj, unsigned int argc, jsval *argv, jsval *rval)
{
    long a, b, sum;

    // Make sure the right number of arguments were passed in.
    if (argc != 2)
        return JS_FALSE;

    // Convert the two arguments from jsvals to longs.
    if (JS_ValueToInteger(cx, argv[0], &a) == JS_FALSE ||
        JS_ValueToInteger(cx, argv[1], &b) == JS_FALSE)
        return JS_FALSE;

    /* Perform the actual work. */
    sum = a + b;

    /* Package the return value as a jsval. */
    *rval = JS_IntegerToValue(sum);

    /* Indicate success. */
    return JS_TRUE;
}
```

Nadat u deze code hebt geschreven, maakt u het DLL-bestand of de gedeelde bibliotheek en sla u deze op in de map Configuration/External Libraries (zie “[C-functies integreren](#)” op pagina 540). Maak vervolgens een JSFL-bestand met de volgende code en sla dit bestand op in de map Configuration/Commands (zie “[JSFL-bestanden opslaan](#)” op pagina 2).

```
// JSFL file to run C function defined above.
var a = 5;
var b = 10;
var sum = Sample.computeSum(a, b);
fl.trace("The sum of " + a + " and " + b + " is " + sum );
```

U kunt de in het DLL-bestand gedefinieerde functie uitvoeren door Opdrachten > Voorbeeld te selecteren in de Flash-ontwerpomgeving.

## Gegevenstypen

De JavaScript-interpreter definieert de gegevenstypen die in deze sectie worden beschreven.

### **typedef struct JSContext JSContext**

Een aanwijzer naar dit ondoorzichtige gegevenstype wordt doorgegeven aan de C-functie. Sommige functies in de API accepteren deze aanwijzer als een van de argumenten.

### **typedef struct JSObject JSObject**

Een aanwijzer naar dit ondoorzichtige gegevenstype wordt doorgegeven aan de C-functie. Dit gegevenstype vertegenwoordigt een object; dit kan een matrixobject of een andere type object zijn.

### **typedef struct jsval jsval**

Een ondoorzichtige gegevensstructuur die een geheel getal kan bevatten of een aanwijzer naar een tekenreeks, een object of een waarde met een drijvende komma. Sommige functies in de API kunnen de waarden van functieargumenten lezen door de inhoud van een `jsval`-structuur te lezen. Andere functies kunnen worden gebruikt om de retourwaarde van de functie te schrijven door een `jsval`-structuur te schrijven.

### **typedef enum { JS\_FALSE = 0, JS\_TRUE = 1 } JSBool**

Een eenvoudig gegevenstype waarmee een Boole-waarde wordt opgeslagen.

## De C-API

De API voor uitbreidbaarheid in C bestaat uit de functiehandtekening `JSBool (*JSNative)` en de volgende functies:

- `JSBool JS_DefineFunction()`
- `unsigned short *JS_ValueToString()`
- `JSBool JS_ValueToInteger()`
- `JSBool JS_ValueToDouble()`
- `JSBool JS_ValueToBoolean()`
- `JSBool JS_ValueToObject()`
- `JSBool JS_StringToValue()`
- `JSBool JS_DoubleToValue()`
- `JSVal JS_BooleanToValue()`
- `JSVal JS_BytesToValue()`
- `JSVal JS_IntegerToValue()`
- `JSVal JS_ObjectToValue()`
- `unsigned short *JS_ObjectType()`
- `JSObject *JS_NewArrayObject()`



- `long JS_GetArrayLength()`
- `JSBool JS_GetElement()`
- `JSBool JS_SetElement()`
- `JSBool JS_ExecuteScript()`

## **typedef JSBool (\*JSNative)(JSContext \*cx, JSObject \*obj, unsigned int argc, jsval \*argv, jsval \*rval)**

### **Beschrijving**

Methode; beschrijft C-implementaties van JavaScript-functies in de volgende situaties:

- De aanwijzer `cx` is een aanwijzer naar een ondoorzichtige `JSContext` -structuur, die moet worden doorgegeven aan enkele functies in de JavaScript API. Deze variabele omvat de uitvoeringscontext van de interpreter.
- De aanwijzer `obj` is een aanwijzer naar het object met de context waarin het script wordt uitgevoerd. Tijdens het uitvoeren van het script is het trefwoord `this` gelijk aan dit object.
- Het gehele getal `argc` is het aantal argumenten dat wordt doorgegeven aan de functie.
- De aanwijzer `argv` is een aanwijzer naar een matrix van `jsval` -structuren. De matrix heeft als lengte `argc` -elementen.
- De aanwijzer `rval` is een aanwijzer naar een `jsval` -structuur. De retourwaarde van de functie moet naar `*rval` worden geschreven.

De functie retourneert `JS_TRUE` indien succesvol en anders `JS_FALSE` . Als de functie `JS_FALSE` retourneert, wordt het huidige script gestopt en wordt er een foutbericht weergegeven.

## **JSBool JS\_DefineFunction()**

### **Gebruik**

```
JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs)
```

### **Beschrijving**

Methode; registreert een C-functie met de JavaScript-interpreter in Flash. Nadat de functie `JS_DefineFunction()` de C-functie heeft geregistreerd die u in het argument `call` hebt opgegeven, kunt u deze functie oproepen in een JavaScript-script door een verwijzing te maken met de naam die u in het argument `name` hebt opgegeven. Het argument `name` is hoofdlettergevoelig.

Deze functie wordt doorgaans aangeroepen door de functie `MM_Init()` , die tijdens het opstarten door Flash wordt aangeroepen.

### **Argumenten**

```
unsigned short * name, JSNative call, unsigned int nargs
```

- Het argument `name` is de naam van de functie zoals deze in JavaScript wordt weergegeven.
- Het argument `call` is een aanwijzer naar een C-functie. De functie moet een `JSBool` retourneren, wat aangeeft of deze functie is voltooid of mislukt.
- Het argument `nargs` is het aantal argumenten dat de functie verwacht te ontvangen.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## unsigned short \*JS\_ValueToString()

### Gebruik

```
unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength)
```

### Beschrijving

Methode; extraheert een functieargument uit een `jsval`-structuur, converteert dit indien mogelijk naar een tekenreeks en retourneert de geconverteerde waarde naar de aanroepende functie.

**Opmerking:** *Wijzig de geretourneerde bufferaanwijzer niet. Als u dit wel doet, kunnen de gegevensstructuren van de JavaScript-interpretter beschadigd raken. Als u de tekenreeks wilt wijzigen, moet u de tekens naar een andere buffer kopiëren en een nieuwe JavaScript-tekenreeks maken.*

### Argumenten

```
JSContext *cx, jsval v, unsigned int *pLength
```

- Het argument `cx` is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `v` is de `jsval`-structuur waaruit de tekenreeks moet worden geëxtraheerd.
- Het argument `pLength` is een aanwijzer naar een niet-ondertekend geheel getal. Deze functie stelt `*pLength` in op de lengte van de tekenreeks in bytes.

### Geretourneerde waarde

Een aanwijzer die naar een tekenreeks met een null-waarde als einde wijst indien de functie is voltooid, of naar een null-waarde als de functie is mislukt. Deze tekenreeks mag na voltooiing niet worden vrijgegeven door de aanroeprountine.

## JSBool JS\_ValueToInteger()

### Gebruik

```
JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp);
```

### Beschrijving

Methode; extraheert een functieargument uit een `jsval`-structuur, converteert dit indien mogelijk naar een geheel getal en retourneert de geconverteerde waarde naar de aanroepende functie.

### Argumenten

```
JSContext *cx, jsval v, long *lp
```

- Het argument `cx` is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `v` is de `jsval`-structuur waaruit het gehele getal moet worden geëxtraheerd.
- Het argument `lp` is een aanwijzer naar een 4-byte functie. Deze functie slaat de geconverteerde waarde op in `*lp`.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSBool JS\_ValueToDouble()

### Gebruik

```
JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp);
```

### Beschrijving

Methode; extraheert een functieargument uit een `jsval`-structuur, converteert dit indien mogelijk naar een dubbele waarde en retourneert de geconverteerde waarde naar de aanroepende functie.

### Argumenten

```
JSContext *cx, jsval v, double *dp
```

- Het argument `cx` is een ondoorzichtige `JSContext`-aanwijzer die is doorgegeven aan de JavaScript-functie.
- Het argument `v` is de `jsval`-structuur waaruit de dubbele waarde moet worden geëxtraheerd.
- Het argument `dp` is een aanwijzer naar een 8-byte functie. Deze functie slaat de geconverteerde waarde op in `*dp`.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSBool JS\_ValueToBoolean()

### Gebruik

```
JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp);
```

### Beschrijving

Methode; extraheert een functieargument uit een `jsval`-structuur, converteert dit indien mogelijk naar een Boole-waarde en retourneert de geconverteerde waarde naar de aanroepende functie.

### Argumenten

```
JSContext *cx, jsval v, JSBool *bp
```

- Het argument `cx` is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `v` is de `jsval`-structuur waaruit de Boole-waarde moet worden geëxtraheerd.
- Het argument `bp` is een aanwijzer naar een `JSBool` Boole-waarde. Deze functie slaat de geconverteerde waarde op in `*bp`.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSBool JS\_ValueToObject()

### Gebruik

```
JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op);
```

### Beschrijving

Methode; extraheert een functieargument uit een `jsval` -structuur, converteert dit indien mogelijk naar een object en retourneert de geconverteerde waarde naar de aanroepende functie. Als het object een matrix is, gebruikt u `JS_GetArrayLength()` en `JS_GetElement()` om de inhoud te lezen.

### Argumenten

`JSContext *cx`, `jsval v`, `JLObject **op`

- Het argument `cx` is een ondoorzichtige `JSContext` -aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `v` is de `jsval` -structuur waaruit het object moet worden geëxtraheerd.
- Het argument `op` is een aanwijzer naar een `JLObject` -aanwijzer. Deze functie slaat de geconverteerde waarde op in `*op`.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSBool JS\_StringToValue()

### Gebruik

```
JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

### Beschrijving

Methode; slaat een geretourneerde tekenreeks op in een `jsval` -structuur. Er wordt een nieuw JavaScript-tekenreeksobject toegewezen.

### Argumenten

`JSContext *cx`, `unsigned short * bytes`, `size_t sz`, `jsval *vp`

- Het argument `cx` is een ondoorzichtige `JSContext` -aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `bytes` is de tekenreeks die moet worden opgeslagen in de `jsval` -structuur. De tekenreeksgegevens worden gekopieerd, dus de aanroepende functie moet de tekenreeks vrijmaken als deze niet nodig is. Als de grootte van de tekenreeks niet is gespecificeerd (zie het argument `sz`), moet de tekenreeks eindigen met een null-waarde.
- Het argument `sz` is de grootte van de tekenreeks in bytes. Als `sz` 0 is, wordt de lengte van de tekenreeks met een null-waarde als einde automatisch berekend.
- Het argument `vp` is een aanwijzer naar de `jsval` -structuur waarnaar de inhoud van de tekenreeks moet worden gekopieerd.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSBool JS\_DoubleToValue()

### Gebruik

```
JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp);
```

### Beschrijving

Methode; slaat een geretourneerd getal met een drijvende komma op in een `j sval` -structuur.

### Argumenten

`J SContext *cx`, `double dv`, `j sval *vp`

- Het argument `cx` is een ondoorzichtige `J SContext` -aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `dv` is een 8-byte getal met een drijvende komma.
- Het argument `vp` is een aanwijzer naar de `j sval` -structuur waarnaar de inhoud van de dubbele waarde moet worden gekopieerd.

### Geretourneerde waarde

Een Boole-waarde: `J S_TRUE` geeft aan dat de functie is voltooid; `J S_FALSE` geeft aan dat de functie is mislukt.

## J SVal JS\_BooleanToValue()

### Gebruik

```
j sval JS_BooleanToValue(JSBool bv);
```

### Beschrijving

Methode; slaat een geretourneerde Boole-waarde op in een `j sval` -structuur.

### Argumenten

`J SBool bv`

- Het argument `bv` is een Boole-waarde: `J S_TRUE` geeft aan dat de functie is voltooid; `J S_FALSE` geeft aan dat de functie is mislukt.

### Geretourneerde waarde

Een `J SVal` -structuur die de Boole-waarde bevat die als een argument aan de functie wordt doorgegeven.

## J SVal JS\_BytesToValue()

### Gebruik

```
J SBool JS_BytesToValue(JSContext *cx, unsigned short *bytes, uint sz, j sval *vp);
```

### Beschrijving

Methode; converteert bytes naar een JavaScript-waarde.

### Argumenten

`J SContext *cx`, `unsignedshort *bytes`, `uintsz`, `j sval *vp`

- Het argument `cx` is de JavaScript-context.
- Het argument `bytes` is de tekenreeks met bytes die naar een JavaScript-object moet worden geconverteerd.
- Het argument `sz` is het aantal bytes dat moet worden geconverteerd.
- Het argument `vp` is de JavaScript-waarde.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSVal JS\_IntegerToValue()

### Gebruik

```
jsval JS_IntegerToValue(long lv);
```

### Beschrijving

Methode; converteert een lang geheel getal naar een `JSVal`-structuur.

### Argumenten

*lv*

Het argument *lv* is het lange gehele getal dat u wilt converteren naar een `jsval`-structuur.

### Geretourneerde waarde

Een `JSVal`-structuur die het gehele getal bevat dat als een argument aan de functie is doorgegeven.

## JSVal JS\_ObjectToValue()

### Gebruik

```
jsval JS_ObjectToValue(JSObject *obj);
```

### Beschrijving

Methode; slaat een geretourneerd object op in een `JSVal`-structuur. Gebruik `JS_NewArrayObject()` om een matrixobject te maken en gebruik `JS_SetElement()` om de inhoud ervan te definiëren.

### Argumenten

`JSObject *obj`

Het argument *obj* is een aanwijzer naar het object `JSObject` dat u naar een `JSVal`-structuur wilt converteren.

### Geretourneerde waarde

Een `JSVal`-structuur die het object bevat dat u als een argument aan de functie hebt doorgegeven.

## unsigned short \*JS\_ObjectType()

### Gebruik

```
unsigned short *JS_ObjectType(JSObject *obj);
```

### Beschrijving

Methode; retourneert de klassennaam van het object aan de hand van een objectverwijzing. Als het object bijvoorbeeld een DOM-object is, retourneert de functie "Document". Als het object een knooppunt in het document is, retourneert de functie "Element". Voor een matrixobject retourneert de functie "Array".

**Opmerking:** *Wijzig de geretourneerde bufferaanwijzer niet. Als u dit wel doet, kunnen de gegevensstructuren van de JavaScript-interpretetor beschadigd raken.*

### Argumenten

`JSObject *obj`

Dit argument wordt doorgaans doorgegeven in en geconverteerd met de functie `JS_ValueToObject()`.

### Geretourneerde waarde

Een aanwijzer naar een tekenreeks met een null-waarde als einde. Deze tekenreeks mag na voltooiing niet worden vrijgegeven door de aanroepende functie.

## JSObject \*JS\_NewArrayObject()

### Gebruik

`JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length [, jsval *v])`

### Beschrijving

Methode; maakt een nieuw object dat de matrix `JSVals` bevat.

### Argumenten

`JSContext *cx`, unsigned int *length*, `jsval *v`

- Het argument *cx* is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument *length* is het aantal elementen dat de matrix kan bevatten.
- Het argument *v* is een optionele aanwijzer naar de `jsvals` die in de matrix moet worden opgeslagen. Als de retourwaarde niet `null` is, is *v* een matrix die *length*-elementen bevat. Als de retourwaarde `null` is, is de initiële inhoud van het matrixobject niet gedefinieerd en kan deze worden ingesteld met de functie `JS_SetElement()`.

### Geretourneerde waarde

Een aanwijzer naar een nieuw matrixobject of de waarde `null` indien de functie mislukt.

## long JS\_GetArrayLength()

### Gebruik

`long JS_GetArrayLength(JSContext *cx, JSObject *obj)`

### Beschrijving

Methode; haalt het aantal elementen in de matrix op aan de hand van een aanwijzer naar een matrixobject.

### Argumenten

`JSContext *cx`, `JSObject *obj`

- Het argument *cx* is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument *obj* is een aanwijzer naar een matrixobject.

### Geretourneerde waarde

Het aantal elementen in de matrix of -1 indien de functie mislukt.

## JSBool JS\_GetElement()

### Gebruik

```
JSBool JS_GetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

### Beschrijving

Methode; leest één element van een matrixobject.

### Argumenten

```
JSContext *cx, JSObject *obj, jsint idx, jsval *vp
```

- Het argument *cx* is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument *obj* is een aanwijzer naar een matrixobject.
- Het argument *idx* is een geheel getal-index voor de matrix. Het eerste element is index 0 en het laatste element is `index ( length -1)`.
- Het argument *vp* is een aanwijzer naar een `jsval`-structuur waarnaar de inhoud van de `jsval`-structuur in de matrix moet worden gekopieerd.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.

## JSBool JS\_SetElement()

### Gebruik

```
JSBool JS_SetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

### Beschrijving

Methode; schrijft één element van een matrixobject.

### Argumenten

```
JSContext *cx, JSObject *obj, jsint idx, jsval *vp
```

- Het argument *cx* is een ondoorzichtige `JSContext`-aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument *obj* is een aanwijzer naar een matrixobject.
- Het argument *idx* is een geheel getal-index voor de matrix. Het eerste element is index 0 en het laatste element is `index ( length -1)`.
- Het argument *vp* is een aanwijzer naar een `jsval`-structuur waarvan de inhoud moet worden gekopieerd naar de `jsval`-structuur in de matrix.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.



## JSBool JS\_ExecuteScript()

### Gebruik

```
JS_ExecuteScript (JSContext *cx, JSObject *obj, unsigned short *script, unsigned int sz, jsval *rval)
```

### Beschrijving

Methode; compileert een JavaScript-tekenreeks en voert deze uit. Als het script een retourwaarde genereert, wordt deze geretourneerd in `*rval`.

### Argumenten

```
JSContext *cx, JSObject *obj, unsigned short *script, unsigned intsz, jsval *rval
```

- Het argument `cx` is een ondoorzichtige `JSContext` -aanwijzer die wordt doorgegeven aan de JavaScript-functie.
- Het argument `obj` is een aanwijzer naar het object met de context waarin het script wordt uitgevoerd. Tijdens het uitvoeren van het script is het trefwoord `this` gelijk aan dit object. Meestal is dit de aanwijzer `JSObject` die wordt doorgegeven aan de JavaScript-functie.
- Het argument `script` is een tekenreeks die JavaScript-code bevat. Als de grootte van de tekenreeks niet is gespecificeerd (zie het argument `sz`), moet de tekenreeks eindigen met een null-waarde.
- Het argument `sz` is de grootte van de tekenreeks in bytes. Als `sz` 0 is, wordt de lengte van de tekenreeks met een null-waarde als einde automatisch berekend.
- Het argument `rval` is een aanwijzer naar een `jsval` -structuur. De retourwaarde van de functie moet worden opgeslagen in `*rval`.

### Geretourneerde waarde

Een Boole-waarde: `JS_TRUE` geeft aan dat de functie is voltooid; `JS_FALSE` geeft aan dat de functie is mislukt.