ADOBE® ILLUSTRATOR® CS6

ADOBE ILLUSTRATOR CS6 SCRIPTING REFERENCE: APPLESCRIPT

Contents

1   **AppleScript Objects**

This chapter provides a complete, alphaetical reference for the objects and commands in the Adobe®
Illustrator® AppleScript dictionary. For each object, the following information is provided:

▶   Elements that can be contained within the object.

▶   Properties of the object, with read-only status, value type, and a description.

▶   Valid commands, with links to sections in Chapter 2, "AppleScript Commands, which describes all
commands in the Illustrator dictionary.

▶   Notes to explain special issues.

▶   Script examples. These example are intended to illustrate concepts; they do not necessarily represent
the best or most efficient way to construct an AppleScript script. Little error checking was done on
them. They assume that the proper context exists for the scripts to execute in; for instance, that there
is a document open or items selected.

For an overview of the Illustrator object model, see *Adobe Illustrator CS6 Scripting Guide*.

application

The Adobe Illustrator application object, which contains all other Illustrator objects.

application elements

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

application object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the application object's value. Always returns `reference`.
browser available	`boolean`	Read-only. If `true`, a web browser is available.
build number	`string`	Read-only. The application's build number.
class	type class	Read-only. The object's class, which is `application`.
color settings	list of Unicode text	Read-only. The list of color-settings files currently available for use.

coordinate system	Valid values:
	   `document coordinate system`
	   `artboard coordinate system`
		Coordinate system currently in use.

current document	`document`	The active (frontmost) document in Illustrator.
default color settings	`file specification`	Read-only. The default color-settings file for the current application locale.
default type	type class	Read-only. The default type for the application object's value. Always returns `reference`.
flattener presets	list of Unicode text	Read-only. The list of flattener style names currently available for use.
free memory	`integer`	Read-only. The amount of unused memory (in bytes) within the Illustrator partition.

Property	Value type	What it is
frontmost	boolean	Read-only. If `true`, this is the frontmost (active) application.
locale	string	Read-only. The application's locale.
name	Unicode text	Read-only. The application's name (not related to the filename of the application file). Always returns `"Adobe Illustrator CS6"`.
PDF presets	list of Unicode text	Read-only. The list of preset PDF-options names available for use.
PPDs	list of PPD files	Read-only. The list of PPD files currently available for use. (A document must be open or an error is returned).
print presets	list of Unicode text	Read-only. The list of preset printing-options names available for use.
printers	list of printers	Read-only. The list of installed printers currently available for use. (A document must be open or an error is returned).
properties	record	All the application's properties returned in a single record. Properties that are individually read-only remain so in this record.
scripting version	Unicode text	Read-only. The version of the Scripting plug-in.

Property	Value type	What it is
`selection`	anything	All currently selected objects in the active (frontmost) document.
		Illustrator does not support the `select` command to change the application's current selection. Use `set the selection to` in place of `select`. See the examples below.
		The application's `selection` can be accessed and modified. When there are no selected objects, `selection` contains an empty list, {}. To deselect all objects in the current document, set `selection` to an empty list.
		When there is an active insertion point in the contents of a text frame, `selection` returns a reference to the insertion point. When characters are selected in the contents of a text frame, `selection` returns a reference to the range of text.
`settings`	[Illustrator preferences](#)	Read-only. Preferences for the Illustrator `application`.
`startup presets`	list of Unicode text	Read-only. The list of preset printing-options names available for use.
`tracing presets`	list of Unicode text	Read-only. The list of preset tracing-options names available for use.

`user interaction level`	Valid values:
	`interact with all`
	`interact with local`
	`interact with self`
	`never interact`
		The level of interaction with the user that is allowed when handling script commands. Default: `interact with all`

`version`	Unicode text	Read-only. The version of the Adobe Illustrator application.

application commands

[activate](#)
[convert sample color](#)
[copy](#)
[cut](#)
[do script](#)
[get preset file of](#)
[get scriptable help group](#)
[launch](#)
[paste](#)

```
quit
redraw
```

Select an object

```
-- Select the first object in the document
tell application "Adobe Illustrator"
-- Make sure there is a page item to select
    if (document 1 exists) and (page item 1 of document 1 exists) then
        set the selection to page item 1 of document 1
    end if
end tell
```

Copy and paste a selection

You do not need to make objects part of the selection to act on them. Selection is useful for moving objects to and from the clipboard using the cut, copy and paste commands, which act on the current selection.

Note that Illustrator must be the front-most application when executing commands that involve the clipboard. This example brings Illustrator to the front using AppleScript's activate command.

```
-- Copy current selection to clipboard then paste into a new doc
tell application "Adobe Illustrator"
    -- If Illustrator is not the frontmost application, activate it.
    if not frontmost then activate
    -- Make sure there is a document to copy from
    if (count documents) > 0 then
        set selectedItems to selection of current document
        if selectedItems is not {} then
            copy
            set colorSpace to color space of current document
            make new document with properties {color space:colorSpace}
            paste
        end if
    end if
end tell
```

artboard, artboards

An artboard object or list of artboard objects. An artboard object represents a single artboard in a document. There can be between 1 to 100 artboards in one document. You cannot delete the last artboard in a document.

artboard object properties

Property	Value type	What it is
artboard rectangle	`rect`	Size and position of the artboard.
best type	type class	Read-only. The best type for the artboard object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `artboard`.
container	object reference	Read-only. A reference to the document that contains this artboard.
default type	type class	Read-only. The default type for the artboard object's value. Always returns `reference`.
index	`integer`	Read-only. The index of this artboard.
name	`text`	The unique identifying name of this artboard.
properties	`record`	The properties of this object, returned as a record.
ruler origin	`point`	Ruler origin of the artboard, relative to the top left corner of the artboard.
ruler PAR	`number (double)`	Pixel aspect ratio, used in ruler visualization if the units are pixels. Range: 0.1 to 10.0
show center	`boolean`	Show center mark.
show cross hairs	`boolean`	Show cross hairs.
show safe areas	`boolean`	Show title and action safe areas (for video).

AutoCAD export options

Options for exporting to an AutoCAD drawing, used with the save command.

AutoCAD export options object properties

Property	Value type	What it is
alter paths for appearance	`boolean`	If `true`, alter paths if needed to maintain appearance. Default: `false`

colors	Valid values:
	   `max 8 colors`
	   `max 16 colors`
	   `max 256 colors`
	   `true colors`
		Number of colors to export into AutoCAD file.

convert text to outlines	`boolean`	If `true`, convert text to outlines. Default: `false`

export file format	Valid values:
	   `dxf`
	   `DWG`
		The format to export to. Default: `DWG`

export option	Valid values:
	   `maintain appearance`
	   `maximize editability`
		Whether to preserve appearance or editability during export. Default: `maximize editability`

export selected art only	`boolean`	If `true`, export only selected artwork. Default: `false`

raster format	Valid values:
	   `PNG raster`
	   `JPEG raster`
		Format in which to export raster art.

scale lineweights	`boolean`	If `true`, scale line weights by the same scaling factor as the rest of the drawing. Default: `false`

scale unit	Valid values:
	   `autocad points`
	   `autocad picas`
	   `autocad inches`
	   `autocad millimeters`
	   `autocad centimeters`
	   `autocad pixels`
		Measurement units from which to map.

scale ratio	`number (double)`	Ratio by which to scale output.

AutoCAD version	Valid values:
	   `AutoCAD release 13`
	   `AutoCAD release 14`
	   `AutoCAD release 15`
	   `AutoCAD release 18`
		The release of AutoCAD to export to.

AutoCAD options

Options for opening an AutoCAD drawing, used with the <u>open</u> command.

AutoCAD options object properties

Property	Value type	What it is
`center artwork`	`boolean`	If `true`, center the artwork that is created on the artboard. Default: `true`
`container`	object reference	Read-only. A reference to the document that contains this object.

`global scale option`	Valid values:
	    `original size`
	    `fit artboard`
	    `scale by value`
		How to scale the drawing on import. Default: `fit artboard`

`global scale percent`	`integer`	The value when `global scale option` is `scale by value`, expressed as a percentage. Range: 0.0 to 100.0 Default: 100.0
`merge layers`	`boolean`	If `true`, the layers of the artwork are merged. Default: `false`
`scale lineweights`	`boolean`	If `true`, scale line weights by the same factor as the rest of the drawing. Default: `false`
`scale ratio`	`integer`	The ratio by which to scale while mapping units. Default: 1.0

`scale unit`	Valid values:
	    `autocad points`
	    `autocad picas`
	    `autocad inches`
	    `autocad millimeters`
	    `autocad centimeters`
	    `autocad pixels`
		The unit to map to. Default: `autocad millimeters`

`selected layout name`	Unicode text	Name of the layout in the drawing to import.

brush, brushes

A brush or list of brushes. Brushes are contained in `document` objects. Scripts cannot create new brushes.

brush object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `brush` object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `brush`.
container	object reference	Read-only. A reference to the document that contains this `brush`.
default type	type class	Read-only. The default type for the `brush` object, which is `reference`.
index	integer	Read-only. The index of this `brush`.
name	Unicode text	The name of this `brush`.
properties	record	All properties of this object returned as a record.

brush object commands

[apply](#)
[count](#)
[exists](#)

Apply brushes

```
-- Duplicate the current selection (if it is a single item)
-- and apply each available brush to the new object
tell application "Adobe Illustrator" to ¬
    set selectedItems to selection

-- Check for selection of single non-text object
if class of selectedItems is text or (count items of selectedItems) is not 1 then
    display dialog "Select a single path item before running this script"
else
    tell application "Adobe Illustrator"
        set pathItem to item 1 of selectedItems
        -- Get the item's position and use it to tile the new items below
        set {itemX, itemY} to position of pathItem
        -- Get a list of all brushes and apply each brush to the selectec item
        set brushList to every brush of current document
        -- Get coordinates of upper-left of document
        set docLeft to 0
        set docTop to height of current document
        set brushCount to count items of brushList
        repeat with i from 1 to brushCount
            set aBrush to item i of brushList
            set itemOffset to i * 20 -- use to tile the duplicated items
            -- Duplicate the selected path item, tiling them from the
```

```
            -- upper-left of the document
            set pathRef to duplicate pathItem to beginning of current document ¬
                with properties {position:{docLeft + itemOffset, docTop - itemOffset}}
            -- Must clear the document's selection before applying a brush
            -- since the duplicate above seems to add to it each time through
            set selection of current document to {}
            apply aBrush to pathRef
        end repeat
    end tell
end if
```

character

Specifies the properties of a character. The text contained within text frames in Illustrator can be accessed using the `character`, `insertion point`, `word`, `line`, `paragraph,` and `text` classes. The properties and valid commands for all these classes are similar, but not identical. For example, while `character` has a `kerning` property, the other text classes do not.

character object elements

Elements	Refer to by
`character style`	name, numeric index, range of elements, before/after another element, satisfying a test
`character`	numeric index, range of elements, before/after another element, satisfying a test
`insertion point`	numeric index, range of elements, before/after another element, satisfying a test
`line`	numeric index, range of elements, before/after another element, satisfying a test
`paragraph style`	name, numeric index, range of elements, before/after another element, satisfying a test
`paragraph`	numeric index, range of elements, before/after another element, satisfying a test
`text`	numeric index, range of elements, before/after another element, satisfying a test
`word`	numeric index, range of elements, before/after another element, satisfying a test

character object properties

Property	Value type	What it is
aki left	`real`	The amount of inter-glyph space added to the left side of the glyph in Japanese text (in thousandths of an em).
aki right	`real`	The amount of inter-glyph spacing added to the right side of the glyph in Japanese text (in thousandths of an em).

alignment	Valid values:
	   `bottom`
	   `center`
	   `icf bottom`
	   `icf top`
	   `roman baseline`
	   `top`
		The character alignment type.

Property	Value type	What it is

alternate glyphs	Valid values:
	  `default`
	  `expert`
	  `full width`
	  `half width`
	  `jis78`
	  `jis83`
	  `proportional width`
	  `quarter width`
	  `third width`
	  `traditional`
	  `jis90`
	  `jis04`
		Specifies which kind of alternate glyphs to use.

auto leading	`boolean`	If `true`, use automatic leading.

baseline direction	Valid values:
	  `standard`
	  `Tate Chu Yoko`
	  `vertical rotated`
		The Japanese text baseline direction.

baseline position	Valid values:
	  `normal`
	  `subscript`
	  `superscript`
		The baseline position of text.

baseline shift	`real`	The amount of shift (in points) of the text baseline.
best type	type class	Read-only. The best type for the object's value.

capitalization	Valid values:
	  `all caps`
	  `all small caps`
	  `normal`
	  `small caps`
		Specifies whether the text is normal, all uppercase, all small caps, or a mix of small caps and lowercase.

character offset	`integer`	Offset of the first character.
class	type class	Read-only. The object's class.
connection forms	`boolean`	If `true`, use the OpenType® connection forms.
container	`reference`	Read-only. The object' s container.
contents	Unicode text	The text content.
contextual ligature	`boolean`	If `true`, use the contextual ligature.
default type	type class	Read-only. The default type for the object's value.
discretionary ligature	`boolean`	If `true`, use the discretionary ligature.

Property	Value type	What it is

figure style	Valid values:
	   `default`
	   `proportional`
	   `proportional oldstyle`
	   `tabular`
	   `tabular oldstyle`
		Specifies the figure style to use in an OpenType font.

fill color	[color info](#)	The color of the text fill.
fractions	`boolean`	If `true`, use OpenType fractions.
horizontal scale	`real`	The horizontal scaling factor for the character.
index	`integer`	Read-only. The index of this instance of the object.
italics	`boolean`	If `true`, the Japanese OpenType supports italics.
kerning	`integer`	Controls the spacing between two characters, in thousandths of an em space.

kerning method	Valid values:
	   `none`
	   `Auto`
	   `Optical`
	   `metricsromanonly`
		The type of automatic kerning method to use.

Property	Value type	What it is

`language`	Valid values:
	  Bokmal Norwegian
	  Brazillian Portuguese
	  Bulgarian
	  Canadian French
	  Catalan
	  Chinese
	  Czech
	  Danish
	  Dutch
	  Dutch 2005 Reform
	  English
	  Finnish
	  German 2006 Reform
	  Greek
	  Hungarian
	  Icelandic
	  Italian
	  Japanese
	  Nynorsk Norwegian
	  old German
	  Polish
	  Romanian
	  Russian
	  Spanish
	  Serbian
	  standard French
	  standard German
	  standard Portuguese
	  Swedish
	  Swiss German
	  Swiss German 2006 Reform
	  Turkish
	  UK English
	  Ukranian
		The language.

`leading`	real	The amount of space between two lines of text, in points.
`length`	integer	The length (in characters). Minimum: 0
`ligature`	boolean	If `true`, use the ligature.
`no break`	boolean	If `true`, no break is allowed.
`ordinals`	boolean	If `true`, use the OpenType ordinals.
`ornaments`	boolean	If `true`, use the OpenType ornaments.
`overprint fill`	boolean	If `true`, overprint the fill of the text.
`overprint stroke`	boolean	If `true`, overprinting of the stroke of the text is allowed.
`properties`	record	All properties of this object returned as a record.

Property	Value type	What it is
proportional metrics	boolean	If `true`, Japanese OpenType supports proportional fonts.
rotation	real	The character rotation angle in degrees.
selection	<u>text</u> or list of <u>text</u>	Read-only. The selected text.
size	real	The font size in points.
story	story	Read-only. The story that contains the object.
strike through	boolean	If `true`, characters use strike-through style.
stroke color	<u>color info</u>	The color of the text stroke.
stroke weight	real	Line width of stroke.
stylistic alternates	boolean	If `true`, use OpenType stylistic alternates.
swash	boolean	If `true`, use the OpenType swash character.
TCY horizontal	integer	The Tate-Chu-Yoko horizontal adjustment in points.
TCY vertical	integer	The Tate-Chu-Yoko vertical adjustment in points.
text font	text font	The text font.
titling	boolean	If `true`, use the OpenType titling alternates.
tracking	integer	The tracking or range kerning amount in thousandths of an em.
Tsume	real	The percentage of space reduction around a Japanese character.
underline	boolean	If `true`, characters use underline style.
vertical scale	real	Character vertical scaling factor, expressed as a percentage (100 is 100%).
warichu characters after break	long	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu characters before break	long	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.

Property	Value type	What it is
`warichu enabled`	`boolean`	If `true`, Wari-Chu is enabled.
`warichu gap`	`integer`	The Wari-Chu line gap in points.

`warichu justification`	Valid values:
	  `auto justify`
	  `center`
	  `full justify last line center`
	  `full justify`
	  `full justify last line left`
	  `full justify last line right`
	  `left`
	  `right`
		The Wari-Chu justification.

`warichu lines`	`integer`	The number of Wari-Chu (multiple text lines fit into a space meant for one) lines.
`warichu scale`	`real`	The Wari-Chu scale.

character object commands

[apply character style](#)
[change case](#)
[count](#)
[delete](#)
[deselect](#)
[duplicate](#)
[exists](#)
[make](#)
[move](#)
[select](#)

Make selected text superscript

```
-- Make the currently selected text superscript
tell application "Adobe Illustrator"
   -- Make sure one or more characters of text are selected
   set selectedText to selection of current document
   if class of selectedText is text or ¬
      class of selectedText is character then
      -- Adjust the properties of the selected text to superscript it
      set fontSize to size of selectedText
      set fontBaseline to baseline shift of selectedText
      set properties of selectedText to ¬
         {size:fontSize / 2, baseline shift:fontBaseline + (fontSize / 2)}
   end if
end tell
```

Stretch characters

This example demonstrates how to use character properties to create unique effects from a script.

```
--Distort every character in the first text frame of a document
--by decreasing the horizontal scaling of each character to the midpoint
--then increasing from the mid point to the end (a smaller value here
--means more difference between largest and smallest horizontal
--scaling of the characters)
tell application "Adobe Illustrator"
    -- Is there is a document and a text frame to work with
    if (exists text frame 1 of current document) then
        -- Make sure the text frame contains some text
        set textframe to first text frame of current document
        if textframe is not "" then -- contains some text
            -- Gather info needed to calculate the scale factor
            set characterCount to count characters in textframe
            set factor to (characterCount + 1) / 2
            -- Iterate over each character, changing its horizontal scale
            repeat with i from 1 to characterCount
                set hScaling to (factor - i) / factor
                if hScaling < 0 then set hScaling to -hScaling
                set widthScale to 100 + 100 * hScaling
                set horizontal scale of character i of text frame 1 of document 1 to
widthScale
            end repeat
        end if
    end if
end tell
```

character style, character styles

A named style that specifies character attributes.

NOTE: Character attributes do not have default values, and are undefined until explicitly set.

character style object properties

Property	Value type	What it is
`aki left`	real	The left aki (in thousandths of an em).
`aki right`	real	The right aki (in thousandths of an em).

`alignment`	Valid values:
	    bottom
	    center
	    icf bottom
	    icf top
	    roman baseline
	    top
		The character alignment type.

`alternate glyphs`	Valid values:
	    default
	    Traditional
	    Expert
	    jis78
	    jis83
	    half width
	    third width
	    quarter width
	    full width
	    proportional width
	    jis90
	    jis04
		The alternate glyphs form.

`alternate ligature`	boolean	If `true`, use the alternate ligature.
`auto leading`	boolean	If `true`, use automatic leading.

`baseline direction`	Valid values:
	    standard
	    Tate Chu Yoko
	    vertical rotated
		The Japanese text baseline direction.

`baseline position`	Valid values:
	    normal
	    subscript
	    superscript
		The baseline position of text.

`baseline shift`	real	The amount of shift (in points) of the text baseline.
`best type`	type class	Read-only. The best type for the object's value.

Property	Value type	What it is

capitalization	Valid values:
	  `all caps`
	  `all small caps`
	  `normal`
	  `small caps`
		The case of the text.

class	type class	Read-only. The object's class.
connection forms	`boolean`	If `true`, use the OpenType connection forms.
contextual ligature	`boolean`	If `true`, use the contextual ligature.
container	`reference`	Read-only. The object's container.
default type	type class	Read-only. The default type for the object's value.
discretionary ligature	`boolean`	If `true`, use the discretionary ligature.

figure style	Valid values:
	  `default`
	  `proportional`
	  `proportional oldstyle`
	  `tabular`
	  `tabular oldstyle`
		Specifies which figure style to use in the OpenType font.

fill color	[color info](#)	The color of the text fill.
fractions	`boolean`	If `true`, use the OpenType fractions.
horizontal scale	`real`	Character horizontal scaling factor expressed as a percentage (100 = 100%).
index	`integer`	Read-only. The index of this instance of the object.
italics	`boolean`	If `true`, the Japanese OpenType supports italics.

kerning method	Valid values:
	  `auto`
	  `none`
	  `optical`
	  `metricsromanonly`
		The automatic kerning method to use.

Property	Value type	What it is

language	Valid values:
	  Bokmal Norwegian
	  Brazillian Portuguese
	  Bulgarian
	  Canadian French
	  Catalan
	  Chinese
	  Czech
	  Danish
	  Dutch
	  Dutch 2005 Reform
	  English
	  Finnish
	  German 2006 Reform
	  Greek
	  Hungarian
	  Icelandic
	  Italian
	  Japanese
	  Nynorsk Norwegian
	  old German
	  Polish
	  Romanian
	  Russian
	  Spanish
	  Serbian
	  standard French
	  standard German
	  standard Portuguese
	  Swedish
	  Swiss German
	  Swiss German 2006 Reform
	  Turkish
	  UK English
	  Ukranian
		The language.

leading	real	The amount of space between two lines of text, in points.
ligature	boolean	If `true`, use the ligature.
name	Unicode text	The character style's name.

OpenType position	Valid values:
	  default
	  denominator
	  numerator
	  subscript
	  superscript
		The OpenType font baseline position.

ordinals	boolean	If `true`, use the OpenType ordinals.
ornaments	boolean	If `true`, use the OpenType ornaments.
overprint fill	boolean	If `true`, the fill of the text should be overprinted.

Property	Value type	What it is
overprint stroke	`boolean`	If `true`, the stroke of the text should be overprinted.
properties	`record`	All properties of this object returned as a record.
proportional metrics	`boolean`	If `true`, the Japanese OpenType font supports proportional glyphs.
rotation	`real`	The character rotation angle in degrees.
size	`real`	The font size in points.
strike through	`boolean`	If `true`, characters use strike-through style.
stroke color	[color info](#)	The color of the text stroke.
stroke weight	`real`	The line width of the stroke.
stylistic alternates	`boolean`	If `true`, use the OpenType stylistic alternates.
swash	`boolean`	If `true`, use the OpenType swash glyph.
TCY horizontal	`integer`	The Tate-Chu-Yoko horizontal adjustment in points.
TCY vertical	`integer`	The Tate-Chu-Yoko vertical adjustment in points.
text font	`text`	The text font.
titling	`boolean`	If `true`, use the OpenType titling alternates.
tracking	`integer`	The tracking or range kerning amount in thousands of an em.
Tsume	`real`	The percentage of space reduction around a Japanese character (100 = 100%).
underline	`boolean`	If `true`, characters use underline style.
vertical scale	`real`	The character vertical scaling factor expressed as a percentage (100 = 100%).
warichu characters after break	`integer`	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu characters before break	`integer`	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu enabled	`boolean`	If `true`, Wari-Chu is enabled.

Property	Value type	What it is
`warichu gap`	`integer`	The Wari-Chu line gap.

`warichu justification`	Valid values:
	`auto justify`
	`center`
	`full justify last line center`
	`full justify`
	`full justify last line left`
	`full justify last line right`
	`left`
	`right`
		The Wari-Chu justification.

`warichu lines`	`integer`	The number of Wari-Chu (multiple text lines fit into a space meant for one) lines.
`warichu scale`	`real`	The Wari-Chu scale.

Character styles

```
-- Create a document with text frames containing text, then create and apply the same
character style to all
tell application "Adobe Illustrator"
    activate
    make new document
    make new text frame in document 1 with properties {name:"text 1",
contents:"Scripting is fun!", position:{50, 100}}
    make new text frame in document 1 with properties {name:"text 2",
contents:"Scripting is easy!", position:{100, 200}}
    make new text frame in document 1 with properties {name:"text 3", contents:"Everyone
should script!", position:{150, 300}}
    make new character style in document 1 with properties {name:"Big Red"}
    set the size of character style "Big Red" of document 1 to 40
    set the tracking of character style "Big Red" of document 1 to -50
    set the capitalization of character style "Big Red" of document 1 to all caps
    set the fill color of character style "Big Red" of document 1 to {class:RGB color
info, red:255, green:0, blue:0}
    -- 'apply character style' is the event.
    -- 'character style "Big Red" of document 1' is the style applied.
    -- note that character styles must be applied to text ranges.
    apply character style character style "Big Red" of document 1 to the text range of
text frame "text 1" of document 1
    apply character style character style "Big Red" of document 1 to the text range of
text frame "text 2" of document 1
    apply character style character style "Big Red" of document 1 to the text range of
text frame "text 3" of document 1
end tell
```

CMYK color info

A CMYK color specification, used to specify a CMYK color where a `color info` object is required. This class contains the color component values of a CMYK color. Use it to specify and get color information from an Illustrator document or page items.

If the `color space` of a document is RGB and you specify the color value for a page item in that document using `CMYK color info`, Illustrator translates the CMYK color specification into an RGB color specification. The same thing happens if the document's color space is CMYK and you specify colors using `RGB color info`. Since this translation can cause information loss you should specify colors using the color info class that matches the document's color space.

CMYK color info object properties

This class inherits all properties from the <u>color info</u> class.

Property	Value type	What it is
cyan	real	The cyan color value. Range: 0.0 to 100.0. Default: 0.0.
magenta	real	The magenta color value. Range: 0.0 to 100.0. Default: 0.0.
yellow	real	The yellow color value. Range: 0.0 to 100.0. Default: 0.0.
black	real	The black color value. Range: 0.0 to 100.0. Default: 0.0.

Create a color swatch

```
-- Make a new CYMK color swatch in the current document
tell application "Adobe Illustrator"
    if not (exists swatch "Our CMYK Swatch" in current document) then
        set swatchColor to {cyan:50.0, magenta:20.0, yellow:20.0, black:0.0}
        make new swatch at end of current document with properties ¬
            {name:"Our CMYK Swatch", color:swatchColor}
    end if
end tell
```

color info

An abstract parent class for all color classes used in Illustrator. Subclasses are:

CMYK color info
gradient color info
gray color info
Lab color info
no color info
pattern color info
RGB color info
spot color info

color management options

Specifies the color management options when printing a document with the [print](#) command.

color management options object properties

Property	Value type	What it is

intent	Valid values:
	`absolute colorimetric`
	`perceptual`
	`relative colorimetric`
	`saturation`
		The color management intent type.
		Default: `relative colorimetric`

name	Unicode text	The color management profile name.

profile kind	Valid values:
	`custom profile`
	`oldstyle profile`
	`printer profile`
	`source profile`
		The color management profile mode.
		Default: `source profile`

color separation options

Print color separation options when printing a document with the <u>print</u> command.

color separation options object properties

Property	Value type	What it is
`convert spot colors`	boolean	If `true`, all spot colors are converted to process colors. Default: `false`
`inks`	list of <u>ink</u>	The list of inks for color separation.
`over print black`	boolean	If `true`, black is overprinted. Default: `false`

`separation mode`	Valid values:
	  `composite`
	  `InRIP separation`
	  `host based separation`
		The color separation type. Default: `composite`

compound path item, compound path items

A compound path or list of compound paths. Compound paths are objects that contain two or more paths that are painted so that holes appear where paths overlap.

All paths in a compound path share property values. Therefore, if you set the value of a property of any one of the paths in the compound path, all other path's matching property will be updated to the new value.

Paths contained within a compound path or group in a document are returned as individual paths when a script asks for the paths contained in the document. However, paths contained in a compound path or group are not returned when a script asks for the paths in a layer which contains the compound path or group.

compound path item object elements

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

compound path item object properties

This object class inherits all properties from the `page item` class.

Property	Value type	What it is
properties	record	All properties of this object returned as a record.

compound path item object commands

count
delete
duplicate
exists
make
move
rotate
scale
transform
translate

Get paths

```
-- get paths in a document that are not part of a compound path or group
tell application "Adobe Illustrator"
    set docRef to current document
    set pathItemList to {}
    set layerCount to count layers of docRef

    repeat with i from 1 to layerCount
        set pathItemList to pathItemList & every path item of layer i of docRef
    end repeat
    set selection to pathItemList
end tell
get pathItemList
```

Duplicate and group paths from a compound path

```
-- Create a group containing a set of paths duplicated from the
-- first compound path item of the document
tell application "Adobe Illustrator"
    set pathItemList to every path item of compound path item 1 of current document
    set groupRef to make new group item at beginning of layer 1 of document 1
    duplicate pathItemList to beginning of groupRef
end tell
```

coordinate options

The print coordinate options when printing a document with the [print](#) command.

coordinate options object properties

Property	Value type	What it is
emulsion	boolean	If `true`, flip the artwork horizontally. Default: `false`
fit to page	boolean	If `true`, proportionally scale the artwork to fit on media. Default: `false`
horizontal scale	real	The horizontal scaling factor. 100.0 = 100%. Range: 1.0 to 10000.0. Default: 100.0.

orientation	Valid values:
	    `landscape`
	    `portrait`
	    `reverse landscape`
	    `reverse portrait`
	    `auto rotate`
		The artwork orientation. Default: `portrait`

position	Valid values:
	    `bottom`
	    `bottom left`
	    `bottom right`
	    `center`
	    `left`
	    `right`
	    `top`
	    `top left`
	    `top right`
		The artwork position on media. Default: `center`

tiling	Valid values:
	    `full pages`
	    `imageable areas`
	    `single full page`
		The page tiling mode. Default: `single full page`

vertical scale	real	The vertical scaling factor. 100.0 = 100%. Range: 1.0 to 10000.0. Default: 100.0.

dataset, datasets

An object, or list of objects, that contains variables and their dynamic data.

dataset object properties

Property	Value type	What it is
`best type`	type class	Read-only. The best type for the dataset's value. Always returns `reference`.
`class`	type class	Read-only. The object's class, which is `dataset`.
`container`	object reference	Read-only. A reference to the art item that contains this `data set`.
`default type`	type class	Read-only. The default type for the `data set`. Always returns `reference`.
`index`	integer	Read-only. The index of this `data set` in the art item.
`name`	Unicode text	The name of the `dataset`.
`properties`	record	All properties of this object returned as a record.

dataset object commands

[count](#)
[delete](#)
[display](#)
[exists](#)
[make](#)
[update](#)

Datasets and variables

```
-- Activate Illustrator
-- Make a new document
-- Make two variables, one of kind visibility and the other textual
-- Make a rectangle and a text frame, and attach the respective variables
-- Set the color of the rectangle and the contents of the text frame
-- Make the first dataset
-- Change the contents of the text and the visibility of the rectangle
-- Make the second dataset
-- display the two datasets
tell application "Adobe Illustrator"
    activate
    make new document
    make new variable in document 1 with properties {name:"RecVariable",
kind:visibility}
    make new variable in document 1 with properties {name:"TextVariable", kind:textual}
    make new rectangle in document 1 with properties {name:"Rec1", position:{100, 500},
visibility variable:variable "RecVariable" of document 1}
    make new text frame in document 1 with properties {name:"Text1", position:{100,
550}, content variable:variable "TextVariable" of document 1}
```

```
    set the fill color of page item "Rec1" of document 1 to {class:RGB color info,
red:150, green:255, blue:255}
    set the contents of text frame "Text1" of document 1 to "Now you see me..."
    make new dataset in document 1 with properties {name:"My First Dataset"}
    set hidden of page item "Rec1" of document 1 to true
    set the contents of text frame "Text1" of document 1 to "Now you don't!"
    make new dataset in document 1 with properties {name:"My Second Dataset"}
    repeat 3 times
        delay 1
        display dataset "My First Dataset" of document 1
        delay 1
        display dataset "My Second Dataset" of document 1
    end repeat
end tell
```

document, documents

An Illustrator document or a list of documents. Documents are contained in the `application` object.

The default document settings—those properties starting with the word "default"—are global settings that affect the current document. Be sure to modify these default properties only when a document is open. Note that if you set default properties to desired values before creating new objects, you can streamline your scripts, eliminating the need to specify properties such as `fill color` and `stroked` that have analogous default properties.

A document's `color space`, `height`, and `width` can only be set when the document is created. Once a document is created, these properties cannot be changed.

The frontmost document can be referred to as either `current document` or `document 1`.

document object elements

Element	Refer to by
`artboard`	name, index, before/after, range, test
`brush`	name, index, before/after, range, test
`character style`	name, index, before/after, range, test
`compound path item`	name, index, before/after, range, test
`dataset`	name, index, before/after, range, test
`gradient`	name, index, before/after, range, test
`graph item`	name, index, before/after, range, test
`graphic style`	name, index, before/after, range, test
`group item`	name, index, before/after, range, test
`layer`	name, index, before/after, range, test
`legacy text item`	name, index, before/after, range, test
`mesh item`	name, index, before/after, range, test
`non native item`	name, index, before/after, range, test
`page item`	name, index, before/after, range, test
`paragraph style`	name, index, before/after, range, test
`path item`	name, index, before/after, range, test
`pattern`	name, index, before/after, range, test
`placed item`	name, index, before/after, range, test
`plugin item`	name, index, before/after, range, test
`raster item`	name, index, before/after, range, test

Element	Refer to by
spot	name, index, before/after, range, test
story	index, before/after, range, test
swatch	name, index, before/after, range, test
swatch group	name, index, before/after, range, test
symbol	name, index, before/after, range, test
symbol item	name, index, before/after, range, test
tag	name, index, before/after, range, test
text frame	name, index, before/after, range, test
variable	name, index, before/after, range, test
view	index, before/after, range, test

document object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the document object's value. Always returns reference.
class	type class	Read-only. The object's class, which is document.
color space	Valid values: RGB CMYK	Read-only. The color specification system to use for this document's color space.
crop marks	rectangle	The boundary of the document's cropping box for output.
crop style	Valid values: standard Japanese style	The style of the document's cropping box.
current dataset	dataset	The currently active dataset.
current layer	layer	The active layer in the document.
current view	view	Read-only. The document's current view.
default fill color	color info	The color to fill new paths if default filled is true.
default fill overprint	boolean	If true, the art beneath a filled object should be overprinted by default.
default filled	boolean	If true, a new path should be filled.
default stroke cap	Valid values: butted rounded projecting	Default type of line capping for paths created.

Property	Value type	What it is
`default stroke color`	color info	The stroke color for new paths if default stroked is `true`.
`default stroke dash offset`	`real`	The default distance into the dash pattern at which the pattern should be started for new paths.
`default stroke dashes`	list of `real` numbers	Default lengths for dashes and gaps in dashed lines, starting with the first dash length, followed by the first gap length, and so on. Set to an empty list, `{}`, for a solid line.

`default stroke join`	Valid values:
	`mitered`
	`rounded`
	`beveled`
		Default type of joints in new paths.

`default stroke miter limit`	`real`	When `default stroke join` is set to `mitered`, this property specifies when the join will be converted to beveled (squared-off) by default. The default miter limit of 4 means that when the length of the point reaches four times the stroke weight, the join switches from a miter join to a bevel join. Values: 1 to 500. 1 specifies a bevel join.
`default stroke overprint`	`boolean`	If `true`, the art beneath a stroked object should be overprinted by default.
`default stroke width`	`real`	Default width of stroke for new paths.
`default stroked`	`boolean`	If `true`, new paths should be stroked.
`default type`	type class	Read-only. The default type for the document object's value. Always returns `reference`.
`file path`	file specification	Read-only. The file associated with the document, which includes the complete path to the file.
`geometric bounds`	rectangle	Read-only. The object's bounds excluding the stroke width.
`height`	`real`	Read-only. The height of the document, calculated from the geometric bounds.
`index`	`integer`	Read-only. The position of this document in the stacking order of all open documents. The current (frontmost) document is always `document 1`.
`inks`	list of ink	Read-only. The list of inks in this document.
`Kinsoku set`	list of Unicode text	Read-only. The Kinsoku set of characters that cannot begin or end a line of Japanese text.
`modified`	`boolean`	If `true`, the document has been modified since the last save.

Property	Value type	What it is
`Mojikumi set`	list of Unicode text	Read-only. A list of names of predefined Mojikumi sets which specify the spacing for the layout and composition of Japanese text.
`name`	Unicode text	Read-only. The document's name (not the complete file path to the document).
`output resolution`	`real`	Read-only. The current output resolution for the document in dots per inch (dpi).
`page origin`	`list`	The zero-point of the page in the document without margins, relative to the overall height and width.
`print tiles`	`boolean`	Read-only. If `true`, this document should print as tiled output.
`properties`	`record`	All document's properties returned in a single record. Properties that are individually read-only remain so in this record.
`raster effect settings`	[raster effect options](#)	The document's raster effect settings.
`ruler origin`	`list`	The zero-point of the rulers in the document relative to the bottom left of the document.

`ruler units`	Valid values:
	`unknown`
	`inches`
	`centimeters`
	`points`
	`picas`
	`millimeters`
	`qs`
	`pixels`
		Read-only. The default units for the rulers in the document.

`selection`	list of object references	The list of references to the objects in this document's current selection.
`show placed images`	`boolean`	Read-only. If `true`, the placed images should be displayed in the document.
`split long paths`	`boolean`	Read-only. If `true`, long paths should be split when printing.
`stationery`	`boolean`	Read-only. If `true`, the document should be saved as a stationery file.
`tile full pages`	`boolean`	Read-only. If `true`, full pages should be tiled when printing this document.
`use default screen`	`boolean`	Read-only. If `true`, use the printer's default screen when printing this document.
`variables locked`	`boolean`	If `true`, the variables are locked.

Property	Value type	What it is
`visible bounds`	rectangle	Read-only. The object's visible bounds, including stroke width of any objects in the illustration.
`width`	`real`	Read-only. The width of this document, calculated from the geometric bounds.
`XMP string`	Unicode text	The XMP metadata packet associated with this document.

document object commands

capture
close
convert coordinate
count
delete
duplicate
exists
export
export PDF preset
export perspective grid preset
export print preset
export variables
fit artboard to selected art
get
get perspective active plane
hide perspective grid
import character styles
import paragraph styles
import PDF preset
import perspective grid preset
import print preset
import variables
make
open
print
rasterize
save
select objects on active artboard
select perspective preset
set perspective active plane
show perspective grid

Make sure a document is open

```
-- Check to make sure a document is open in Illustrator
-- before setting the application's default stroke width to 8 points
tell application "Adobe Illustrator"
    if not (document 1 exists) then
        make new document with properties {color space:CMYK, width:100.0, height:50.0}
    end if
    set the default stroke width of document 1 to 8.0
end tell
```

Make a new document

```
-- Creates 2 new documents with different default settings
-- the RGB document has the default fill and a 4.0 pt stroke
-- the CMYK document has no fill and a dashed stroke width of 8.0 pt
tell application "Adobe Illustrator"
    set rgbDocRef to make new document with properties {color space:RGB}
    set properties of rgbDocRef to {default filled:true ¬
        , default stroked:true ¬
        , default stroke width:4.0}
    set rgbPropertyRef to properties of current document
    set cmykDocRef to make new document with properties {color space:CMYK}
    set properties of cmykDocRef to {default filled:false ¬
        , default stroked:true ¬
        , default stroke width:8.0 ¬
        , default stroke dashes:{2.5, 1, 2.5, 1, 2.5, 1}}
    set cmykPropertyRef to properties of current document
end tell
```

Get the file path of a document

This example demonstrates how to use document properties in other applications. In this case, the script uses the `file path` property of the active document to open the folder containing the Illustrator document in the Finder.

```
-- Reveal and select a document's file icon in the Finder
tell application "Adobe Illustrator"
    set filepath to file path of current document
end tell
tell application "Finder"
    activate
    reveal filepath
end tell
```

document preset

A preset document template to use when creating a new document. See the [add document](#) command.

document preset properties

Property	Value type	What it is

artboardLayout	Valid values:
	  grid by row
	  grid by column
	  row
	  column
	  rl grid by row
	  rl grid by col
	  rl row
		The layout of artboards in the new document. Default: `grid by row`

artboardRowsOrCols	long	The number of rows (for rows layout) or columns (for column layout) of artboards. Range: 1 to (`numArtboards` - 1) or 1 for single row or column layouts. Default: 1
artboardSpacing	double	The spacing between artboards in the new document. Default: 20.0

color mode	Valid values:
	  CMYK
	  RGB
		The color space for the new document. Default: `CMYK`

document units	Valid values:
	  centimeters
	  inches
	  millimeters
	  picas
	  points
	  qs
	  pixels
	  unknown
		The ruler units for the new document. Default: `points`

height	real	The height in document points. Default: 792.0
numArtboards	long	The number of artboards for the new document. Range: 1 to 100. Default: 1

preview mode	Valid values:
	  default preview
	  pixel preview
	  overprint preview
		The preview mode for the new document.
		Default: `default preview`

raster resolution	Valid values:
	  screen resolution
	  medium resolution
	  high resolution
		The raster resolution for the new document.
		Default: `screen resolution`

Property	Value type	What it is
title	Unicode text	The document title. Default: `Untitled`

transparency grid	Valid values:
	`hide transparency grids`
	`light color transparency grids`
	`medium color transparency grids`
	`dark color transparency grids`
	`red color transparency grids`
	`orange transparency grids`
	`green transparency grids`
	`blue transparency grids`
	`purple transparency grids`
		The transparency grid color for the new document.
		Default: `hide transparency grids`

width	real	The width in document points. Default: 612.0

ellipse

Used to create an elliptical path in an Illustrator document. This object is available only in the context of a make command, which creates an instance of the path item class. This special class allows you to quickly create complex path items. Properties associated with path items, such as fill color and note, can also be specified at the time of creation.

ellipse object properties

Property	Value type	What it is
bounds	list of points	Write-once. The bounds of the ellipse.
inscribed	boolean	Write-once. If true, the ellipse path should be inscribed (drawn inside the rectangle described by the bounds).
reversed	boolean	Write-once. If true, the ellipse path is reversed. Default: false

ellipse object commands

make

Create ellipses

```
-- Embellish a single selected path item by adding a bright red
-- ellipse to each point on the path

set pEllipseScale to 0.1

tell application "Adobe Illustrator"
    activate
    set selectedItems to selection

    -- A bit of sanity checking
    if (count selectedItems) is not 1 ¬
        or class of selectedItems is text ¬
        or class of item 1 of selectedItems is not path item then

        display dialog "Please select a single path item before running this script"
    else
        set pathItem to item 1 of selectedItems

        -- Set ellipse color based on document color space
        set docColorSpace to color space of current document
        if docColorSpace is RGB then
            set ellipseColor to {red:255.0, green:0.0, blue:0.0}
        else
            set ellipseColor to {cyan:0.0, magenta:100.0, yellow:100.0, black:0.0}
        end if

        -- Gather needed info about the path item to be embellished
        set itemWidth to width of pathItem
        set itemHeight to height of pathItem
        set pathPointList to anchor of every path point of pathItem
```

```
        -- Calculate the position and bounds for each ellipse
        repeat with aPoint in pathPointList
            set {x, y} to aPoint

            set rectLeft to x - (itemWidth * pEllipseScale)
            set rectRight to x + (itemWidth * pEllipseScale)
            set rectTop to y + (itemHeight * pEllipseScale)
            set rectBottom to y - (itemHeight * pEllipseScale)

            set ellipseRect to {rectLeft, rectTop, rectRight, rectBottom}

            make new ellipse at beginning of current document with properties
{bounds:ellipseRect, inscribed:true, reversed:false, stroke color:ellipseColor, fill
color:ellipseColor}
        end repeat
    end if
end tell
```

EPS save options

Options that can be supplied when saving a document as an Illustrator EPS file. See the <u>save</u> command for additional details.

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

EPS save options object properties

Property	Value type	What it is
artboard range	string	Optional. If `save multiple artboards` is `true`, this is considered for multi-asset extraction, which specifies the artboard range. An empty string extracts all the artboards. Default: empty string
CMYK PostScript	boolean	Optional. If `true`, the file should be saved as CMYK PostScript. Default: `false`

compatibility	Valid values:
	`Illustrator 3`
	`Illustrator 8`
	`Illustrator 9`
	`Illustrator 10`
	`Illustrator 11`
	`Illustrator 12`
	`Illustrator 13`
	`Illustrator 14`
	`Illustrator15`
	`Japanese 3`
		Optional. The Illustrator file format version to create. Default: `Illustrator 15`

compatible gradient printing	boolean	Optional. If `true`, create a raster item of the gradient or gradient mesh so that PostScript Level 2 printers can print the object.
embed all fonts	boolean	Optional. If `true`, fonts used in the EPS file should be embedded in the file (version 7 or later). Default: `false`
embed linked files	boolean	Optional. If `true`, linked image files are to be included in the saved document. Default: `false`

flatten output	Valid values:
	`preserve paths`
	`preserve appearance`
		Optional. How transparency should be flattened for file formats before Illustrator 9. Default: `preserve appearance`

included document thumbnails	boolean	Optional. If `true`, the thumbnail image of the EPS artwork should be included. Default: `true`

Property	Value type	What it is

overprint	Valid values:
	`discarded`
	`preserved`
		Optional. The overprint style. Default:
		`preserved`

PostScript	Valid values:
	`level 2`
	`level 3`
		Optional. Specifies the PostScript level to use when saving the file (level 1 is valid for file format version 8 or older). Default: `level 3`

preview	Valid values:
	`none`
	`BW TIFF`
	`color TIFF`
	`transparent color TIFF`
		Specifies the format for the EPS preview image. Default: `color TIFF`

save multiple artboards	boolean	Optional. If `true`, all artboards or range of artboards are saved. Default: `false`

Save EPS files

This handler processes a folder of Illustrator files, saving each as an EPS file with level 2 PostScript and Illustrator CS6 compatibility. The files are save to the folder specified in the `destinationFolder` parameter. Note that the `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destinationFolder is an alias to a folder where the files are to be saved

on ExportFilesAsEPS(fileList, filePath, destinationFolder)
    set destinationPath to destinationFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destinationPath & fileName & ".EPS"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                save current document in file newFilePath as eps ¬
                    with options {class:EPS save options ¬
                    , compatibility:Illustrator 9 ¬
                    , preview:color TIFF ¬
                    , embed linked files:true ¬
                    , include document thumbnails:true ¬
                    , embed all fonts:true ¬
                    , CMYK PostScript:true ¬
                    , PostScript:level 2}
                close current document saving no
            end tell
        end repeat
    end if
end ExportFilesAsEPS
```

Flash export options

You can supply a number of options when exporting a document as Macromedia® Flash™ (SWF). See the export command in the command reference for additional details.

This class is used to define a record containing properties that specify options when exporting a document as a Flash (SWF) file. `Flash export options` can be supplied only in conjunction with the `export` command. It is not possible to get or create a `Flash export options` object.

All properties are optional.

Flash export options object properties

Property	Value type	What it is

art clipping	Valid values:
	    output art bounds
	    output artboard bounds
	    output croprect bounds
		How the arts should be clipped during the output. Default: output art bounds

artboard range	string	If save multiple artboards is true, this is considered for multi-asset extraction, which specifies the artboard range. An empty string extracts all the artboards. Default: empty string
background color	RGB color info	The background color.
background layers	list of layers	Layers to be included as the static background in all exported Flash frames.

blend animation	Valid values:
	    in build
	    in sequence
	    none
		How the blend art objects are animated when exported to Flash frames. Default: none

compressed	boolean	If true, the exported file should be compressed. Default: false
convert text to outlines	boolean	If true, all text should be converted to outlines. Default: false
curve quality	integer	How much curve information should be preserved. Range: 0 to 10. Default: 7
export all symbols	boolean	If true, export all symbols defined in the palette. Default: false

export style	Valid values:
	    Flash file
	    layers to frames
	    layers to files
	    layers to symbols
	    Artboards to Files
		How the Flash file should be created Default: Flash file

Property	Value type	What it is
export version	Valid values: SWF version 1 SWF version 2 SWF version 3 SWF version 4 SWF version 5 SWF version 6 SWF version 7 SWF version 8 SWF version 9	The version of the exported SWF file. Default: SWF version 9
Flash Playback Security	Valid values: flash playback local access flash playback network access	Security access for playback. Default: flash playback local access
frame rate	real	When exporting layers to Flash frames Range: 0.01 to 120.0. Default: 12.0
image format	Valid values: lossless lossy	How the images in the exported file should be compressed. Default: lossless
include metadata	boolean	If true, include minimal XMP metadata in the SWF file. Default: false
JPEG method	Valid values: optimized standard	Specifies which method to use. Default: standard
JPEG quality	integer	Level of compression. Range: 0 to 10, Default: 3
layer order	Valid values: bottom up top down	The order in which layers should be exported to Flash frames. Default: bottom up
looping	boolean	If true, the Flash file should be set to loop when run. Default: false
preserve appearance	boolean	If true, preserve appearance. If false, preserve editability. Default: false
read only	boolean	If true, export as read only file. Default: false
replacing	Valid values: yes no ask	If a file with the same name already exists, should it be replaced. Default: ask
resolution	real	Pixels per inch. Range: 72 to 2400. Default: 72

Property	Value type	What it is
save multiple artboards	boolean	If `true`, all artboards or range of artboards are saved. Default: `false`
textkerning	boolean	If `true`, ignore kerning information in text objects. Default: `false`

flattening options

Specifies transparency flattening options when printing a document with the [print](#) command. These options are used to output artwork that contains transparency into a non-native format.

flattening options object properties

Property	Value type	What it is
`clip complex regions`	boolean	If `true`, complex regions are clipped. Default: `false`
`convert strokes to outlines`	boolean	If `true`, all strokes are converted to outlines. Default: `false`
`convert text to outlines`	boolean	If `true`, all text items are converted to outlines. Default: `false`
`flattening balance`	integer	The flattening balance. Range: 0 to 100; Default: 100
`gradient resolution`	real	The gradient resolution in dots per inch. Range: 1.0 to 9600.0; Default: 300.0
`overprint`	Valid values: discard preserve	Overprint choice. Default: `preserve`
`rasterization resolution`	real	The rasterization resolution in dots per inch. Range: 1.0 to 9600.0. Default: 300.0

Flattening options

```
-- Activate Illustrator
-- Create a variable that holds the flattening options
-- Create a variable that holds the print options
-- Print the document
tell application "Adobe Illustrator"
    activate
    set flatOpts to {class:flattening options, clip complex regions:true, gradient
resolution:360, rasterization resolution:360}
    set printOpts to {class:print options, flattener settings:flatOpts}
    if not (exists document 1) then error "There is no document available to print."
    print document 1 options printOpts
end tell
```

font options

Font options when printing a document with the <u>print</u> command.

font options object properties

Property	Value type	What it is

`download fonts`	Valid values:
	    `complete`
	    `none`
	    `subset`
		The font download mode. Default: `subset`

`font substitution kind`	Valid values:
	    `device substitution`
	    `oblique substitution`
	    `tint substitution`
		The font substitution policy.
		Default: `oblique substitution`

Set font options

```
-- Set the font options to a desired value
-- Print the current document, if available
tell application "Adobe Illustrator"
    activate
    set fontOptions to {class:font options, download fonts:complete, font substitution
kind:device substitution}
    set printOpts to {class:print options, font settings:fontOptions}
    if not (exists document 1) then error "There is no document available to print."
    print document 1 options printOpts
end tell
```

FreeHand options

Options for opening a FreeHand file.

FreeHandFileOptions properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
container	object reference	Read-only. A reference to the document that contains this object.
convert text to outlines	boolean	If `true`, converts all text to outlines. Default: `false`
default type	type class	Read-only. The default type for the object's value.
import single page	boolean	If `true`, imports only the page specified in the `page` property. Default: `true`
page	long	The number of the page to import when opening a multipage document. **NOTE:** Valid only when `import single page` is `true`.
properties	record	All properties of this object returned as a record.

FXG save options

Specifies options which may be supplied when saving a document as an FXG file. All properties are optional.

FXG save options object properties

Property	Value type	What it is
artboard range	string	If `save multiple artboards` is `true`, this is considered for multi-asset extraction, which specifies the artboard range. An empty string extracts all the artboards. Default: empty string

blends policy	Valid values:
	    autoConvert blends
	    rasterize blends
		The policy used by FXG to expand blends. Default: `autoConvert blends`

downsample linked images	boolean	If `true`, linked images are downsampled (at 72 dpi). Default: `false`

filters policy	Valid values:
	    expand filters
	    keep filters editable
	    rasterize filters
		The policy used by FXG to preserve filters. Default: `keep filters editable`

fxg version	Valid values:
	    version 1.0
	    version 2.0
		The version of the FXG file format to create. Default: `version 2.0`

gradients policy	Valid values:
	    autoConvert gradients
	    keep gradients editable
		The policy used by FXG to preserve gradients. Default: `autoConvert gradients`

include unused symbols	boolean	If `true`, unused symbols are included. Default: `false`
preserve editing capabilities	boolean	If `true`, the editing capabilities of FXG are preserved. Default: `true`
save multiple artboards	boolean	If `true`, all artboards or range of artboards are saved. Default: `false`

text policy	Valid values:
	    outline text
	    keep text editable
	    rasterize text
	    autoConvert text
		The policy used by FXG to preserve text. Default: `autoConvert text`

GIF export options

Options that can be supplied when exporting a document as a GIF file. See the export command in the command reference for additional details.

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

GIF export options object properties

Property	Value type	What it is
antialiasing	boolean	If `true`, the resulting image should be anti-aliased. Default: `true`
artboard clipping	boolean	If `true`, the resulting image should be clipped to the artboard. Default: `false`
color count	integer	The number of colors in the exported color table. Range: 2 to 256. Default: 128
color dither	Valid values: none diffusion pattern dither noise	The method used to dither colors. Default: `diffusion`
color reduction	Valid values: selective adaptive perceptual web	The method used to reduce the number of colors in the document. Default: `selective`
dither percent	integer	How much the colors should be dithered. Range: 0 to 100. Default: 88
horizontal scaling	real	The horizontal scaling factor to apply to the resulting image. Range: 0.0 to 100.0. Default: 100.0
information loss	integer	The level of information loss during compression (as a percentage). Range: 0 to 100. Default: 0
interlaced	boolean	If `true`, the resulting image should be interlaced. Default: `false`
matte	boolean	If `true`, the artboard should be matted with a color. Default: `true`
matte color	RGB color info	The color to use when matting the artboard. Default: `white`
saving as HTML	boolean	If `true`, the resulting image is saved with an accompanying HTML file. Default: `false`

Property	Value type	What it is
transparency	boolean	If `true`, the resulting image uses transparency. Default: `true`
vertical scaling	real	The vertical scaling factor to apply to the resulting image. Range: 0.0 to 100.0. Default: 100.0
web snap	integer	How much the color table should be changed to match the Web pallet. Range: 0 to 100, where 100 is the maximum change. Default: 0

Export to GIF

This handler processes all Illustrator files in a specific folder, exporting each as a scaled GIF image. Note that the `class` property is specified in the record to ensure that Illustrator can determine the export option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destinationFolder is an alias to a folder where the files are to be saved

on ExportFilesAsGIF(fileList, filePath, destinationFolder)
    set destinationPath to destinationFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destinationPath & fileName & ".gif"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                export current document to file newFilePath as GIF with options ¬
                    {class:GIF export options ¬
                        , color count:256 ¬
                        , color reduction:adaptive ¬
                        , information loss:0 ¬
                        , color dither:none ¬
                        , dither percent:100 ¬
                        , web snap:0 ¬
                        , transparency:false ¬
                        , interlaced:false ¬
                        , matte:true ¬
                        , matte color:{red:128, green:0, blue:60} ¬
                        , horizontal scaling:50.0 ¬
                        , vertical scaling:50.0 ¬
                        , antialiasing:true ¬
                        , artboard clipping:false ¬
                        , saving as HTML:false}
                close current document saving no
            end tell
        end repeat
    end if
end ExportFilesAsGIF
```

gradient, gradients

A gradient definition or gradient definitions. Gradients are contained in documents. Scripts can create new gradients.

gradient object elements

Element	Refer to by
gradient stop	index, before/after, range, test

gradient object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the gradient object's value. Always returns reference.
class	type class	Read-only. The object's class, which is gradient.
container	object reference	Read-only. A reference to the document that contains this gradient.
default type	type class	Read-only. The default type for the gradient object's value. Always returns reference.
entire gradient	list of gradient stop info	All gradient stops in the gradient.
gradient type	Valid values: linear radial	The type of the gradient.
index	integer	Read-only. The position of this gradient in the application.
name	Unicode text	The gradient's name.
properties	record	All properties of this object returned as a record.

gradient object commands

count
delete
duplicate
exists
make

Create a gradient

```
-- Create a new RGB gradient with three gradient stops
set pGradientName to "RGB Hot Streak"

tell application "Adobe Illustrator"
    if not (exists gradient pGradientName in current document) then
        set newgradient to make new gradient at beginning of current document Â
            with properties {name:pGradientName, gradient type:linear}
        -- Since all new gradients are created with 2 gradient stops,
        -- create another stop for the 3 stop gradient
        make new gradient stop at beginning of newgradient
        set properties of gradient stop 1 of newgradient to Â
            {midpoint:50.0, ramp point:0.0, color:{red:255.0, green:255.0, blue:0.0}}
        set properties of gradient stop 2 of newgradient to Â
            {midpoint:50.0, ramp point:50.0, color:{red:255.0, green:127.0, blue:127.0}}
        set properties of gradient stop 3 of newgradient to Â
            {midpoint:50.0, ramp point:100.0, color:{red:255.0, green:0.0, blue:0.0}}
    end if
end tell
```

gradient color info

A gradient color specification, used to specify the color component values of a gradient color swatch. It is used for specifying and retrieving color information from an Illustrator document or from page items in a document.

gradient color info object properties

This class inherits all properties from the `color info` class.

Property	Value type	What it is
`angle`	`real`	The gradient vector angle (in degrees). Default: 0.0
`gradient`	object reference	A reference to the gradient object that defines the gradient to use in this color definition.
`hilite angle`	`real`	The gradient highlight vector angle in degrees. Default: 0.0
`hilite length`	`real`	The gradient highlight vector length. Default: 0.0
`length`	`real`	The gradient vector length.
`matrix`	`matrix`	An additional transformation matrix to manipulate the gradient path.
`origin`	`fixed point`	The gradient vector origin.

Gradient information

```
-- Set fill color of the first path in the current document
-- to the first gradient in the document
tell application "Adobe Illustrator"
    set the fill color of path item 1 of document 1 to ¬
        {gradient:gradient 1 of document 1}
end tell
```

gradient stop, gradient stops

A gradient stop definition or definitions contained in a specific gradient. A gradient stop is a point on a specific gradient that specifies a color change in the containing gradient.

gradient stop object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the gradient stop object's value. Always returns reference.
class	type class	Read-only. The object's class, which is gradient stop.
color	color info	The color linked to this gradient stop.
container	object reference	Read-only. A reference to the gradient that contains this gradient stop.
default type	type class	Read-only. The default type for the gradient stop object's value. Always returns reference.
index	integer	Read-only. The position of this gradient stop in the gradient.
midpoint	real	The midpoint of the blend between this stop's and the next stop's colors. Range: 13.0 to 87.0
properties	record	All properties of this object returned as a record.
ramp point	real	The location of the color in the gradient. Range: 0.0 to 100.0
stop opacity	double	The opacity value for the gradient stop. Range: 0.0 to 100.0

gradient stop object commands

count
delete
duplicate
exists
make

Reverse colors in a gradient

```
-- This script reverses the colors in the first gradient of the current document
tell application "Adobe Illustrator"
    set gradientRef to gradient 1 of document 1
    -- Get a list of the gradient's colors
    set colorList to color of every gradient stop of gradientRef
    -- Tell AppleScript to reverse the order of the list
    set colorList to reverse of colorList
    -- Iterate over the gradient resetting its colors
    set colorCount to count items in colorList
    repeat with i from 1 to colorCount
        set color of gradient stop i of gradientRef to (item i of colorList)
    end repeat
```

```
    end tell
```

gradient stop info

Gradient stop information of a specific gradient, returned by the `entire gradient` property of a gradient.

The gradient stops for a new gradient can be specified by providing a list of `gradient stop info` records in the `entire gradient` property. The following applies when creating a gradient from a list of gradient stop info records:

▶ A gradient stop's location in the gradient is determined by its `ramp point` value, not the `gradient stop info` record's order in the entire gradient list.

▶ The `midpoint` value of the last `gradient stop info` record in the entire gradient list is not used for the newly created gradient and need not be provided. If it is present, its value must be in the valid range.

gradient stop info object properties

Property	Value type	What it is
`color`	color info	The color linked to this gradient stop.
`midpoint`	real	The midpoint of the blend between this stop's and the next stop's colors. Range: 13.0 to 87.0. Default: 50.0
`ramp point`	real	The location of the color in the gradient as a percentage. Range: 0.0 to 100.0. Default: 0.0
`stop opacity`	Double	The opacity value for the gradient stop. Range: 0.0 to 100.0. Default: 100.0

Gradient stop information

```
-- Create a new CMYK gradient with 4 gradient stops
set pGradientName to "CMYK Circle"
tell application "Adobe Illustrator"
    if not (exists gradient pGradientName in current document) then
        set entireGradient to {{midpoint:50.0, ramp point:0.0 ¬
            , color:{cyan:0.0, magenta:0.0, yellow:0.0, black:100.0}} ¬
            , {midpoint:50.0, ramp point:33.3 ¬
            , color:{cyan:0.0, magenta:0.0, yellow:100.0, black:0.0}} ¬
            , {midpoint:50.0, ramp point:66.7 ¬
            , color:{cyan:0.0, magenta:100.0, yellow:0.0, black:0.0}} ¬
            , {midpoint:50.0, ramp point:100.0 ¬
            , color:{cyan:100.0, magenta:0.0, yellow:0.0, black:0.0}}}
        set gradientRef to make new gradient in current document with properties ¬
            {name:pGradientName, gradient type:radial, entire gradient:entireGradient}
    end if
end tell
```

graph item, graph items

A graph or a list of graphs.

graph item object properties

This object class inherits all properties from the `page item` class.

Property	Value type	What it is
content variable	anything	The content variable to which this `graph item` is bound
		It is not necessary to set the type of the `content variable` before binding. Illustrator automatically sets the type to `graph`.
properties	record	All properties of this object returned as a record.

graph item object commands

count
delete
duplicate
exists
move
rotate
scale
transform
translate

Rotating graph items

```
-- Get every page item whose class is graph item
-- For each graph item, rotate it 90 degrees counter clockwise
tell application "Adobe Illustrator"
    activate
    if not (exists document 1) then error "There is no available document."
    set graphItems to every page item of document 1 whose class is graph item
    if graphItems is {} then error "The document does not contain any graph items."
    repeat with currentGraphItem in graphItems
        rotate currentGraphItem angle 90
    end repeat
end tell
```

graphic style, graphic styles

Defines a set of appearance attributes that you can apply non-destructively to page items. Graphic styles are contained in documents. The graphic styles can be accessed from a script, but cannot be created from a script. You cannot delete default `graphic styles`.

graphic style object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `graphic style` object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `graphic style`.
container	object reference	Read-only. A reference to the document that contains this `graphic style`.
default type	type class	Read-only. The default type for the `graphic style` object, which is `reference`.
index	integer	Read-only. The index of this `graphic style`.
name	Unicode text	The name of this `graphic style`.
properties	record	All properties of this object returned as a record.

graphic style object commands

[apply](#)
[count](#)
[delete](#)
[exists](#)

Applying a graphic style

```
-- Duplicate and group the selected path items, then apply
-- a random graphic style to the items in the new group
tell application "Adobe Illustrator"
    set selectedItems to selection of document 1

    -- Check for empty selection
    if selectedItems is not {} then
        -- Create the new group to contain the duplicated items
        set groupRef to make new group item at document 1
        -- Duplicate the selected items to the new group
        set newItemList to duplicate selectedItems to beginning of groupRef
        -- Get graphic style names for display in the choice list
        set styleIndex to index of every graphic style of document 1
        if (count styleIndex) > 0 then
            -- select a random graphic style
            set chosenStyle to (random number from 1 to (count styleIndex))
            -- The randomly chosen graphic style is applied to the list
            -- of items returned by the duplicate command,
```

```
                    -- rather than to the new group itself, because the
                    -- apply command works on individual path items,
                    -- not groups of items
                    apply graphic style chosenStyle of current document to newItemList
            end if
        end if
end tell
```

gray color info

A grayscale color specification, used to specify a gray color where a `color info` object is required.

This class is used to define a record which contains the tint value of a gray color. It is used for specifying and retrieving color information from an Illustrator document or from page items in a document.

gray color info object properties

This class inherits all properties from the <u>color info</u> class.

Property	Value type	What it is
gray value	real	The tint of the gray. Range: 0.0 (white) to 100.0 (black). Default: 0.0

Creating a gray color swatch

```
-- Create a new gray color swatch (35% black) in the current document
set pSwatchName to "35% Gray Swatch"
tell application "Adobe Illustrator"
    if not (exists swatch pSwatchName in current document) then
        make new swatch at beginning of current document with properties ¬
            {name:pSwatchName, color:{gray value:35.0}}
    end if
end tell
```

group item, group items

A grouped set of art items. Group items can contain all the same page items that a layer can contain, including other nested groups.

Paths contained within a group or compound path in a document are returned as individual paths when a script asks for the paths contained in the document. However, paths contained in a group or compound path are not returned when a script asks for the paths in a layer which contains the group or compound path.

A new group can be created that contains the contents of a vector art file if you provide a file specification to the vector file (EPS or PDF) in the `with data` parameter of the `make` command. The resulting group will be the same object as if the user had placed the file from the user interface using the **File > Place** command with the embed checkbox checked.

group item object elements

Element	Refer to by
compound path item	name, index, before/after, range, test
graph item	name, index, before/after, range, test
group item	name, index, before/after, range, test
legacy text item	name, index, before/after, range, test
mesh item	name, index, before/after, range, test
non native item	name, index, before/after, range, test
page item	name, index, before/after, range, test
path item	name, index, before/after, range, test
placed item	name, index, before/after, range, test
plugin item	name, index, before/after, range, test
raster item	name, index, before/after, range, test
symbol item	name, index, before/after, range, test
text frame	name, index, before/after, range, test

group item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
clipped	boolean	If `true`, the `group item` is clipped to the clipping mask.
properties	record	All properties of this object returned as a record.

group item object commands

count
delete
duplicate
exists
make
move
rotate
scale
transform
translate

Group contents of a vector art file

```
-- Create a new group whose contents will be the contents of a vector art file
-- fileRef is an alias or file reference to the vector file to be placed
on EmbedVectorFile(fileRef)
    tell application "Adobe Illustrator"
        set groupRef to make new group item in document 1 with data fileRef ¬
            with properties {position:{0, 600}}
    end tell
    return groupRef
end EmbedVectorFile

-- Call handler
set fileRef to choose file with prompt "Select vector file to place"
set groupRef to EmbedVectorFile(fileRef)
```

Create path items from a group

This script demonstrates how easily new groups can be created and populated with objects.

```
-- Create a new group, then add rectangles to it using
-- the available placement options
tell application "Adobe Illustrator"
    set groupRef to make new group item in document 1
    set rectRef to make new rectangle at beginning of groupRef with properties Â
        {bounds:{150, 550, 350, 350}, fill color:{blue:255}}
    make new rectangle after rectRef with properties Â
        {bounds:{100, 600, 300, 400}, fill color:{red:255}}
    set rectRef to make new rectangle at end of groupRef with properties Â
        {bounds:{0, 700, 200, 500}, fill color:{green:255}}
    make new rectangle before rectRef with properties Â
        {bounds:{50, 650, 250, 450}, fill color:{black:100}}
end tell
```

Select items not in a group

```
-- Select only the page items in a document that are not part of
-- a group and that are not themselves groups
tell application "Adobe Illustrator"
    -- First deselect everything in the document
    set selection of current document to {}
    if (count page items of current document) > 0 then
        set layerCount to count layers in current document
        repeat with i from 1 to layerCount
            set layerRef to layer i of current document
            if (count page items of layer i of current document) > 0 then
                set selected of (every page item of current document ¬
                    whose container is layerRef ¬
                    and class is not group item) to true
            end if
        end repeat
    end if
end tell
```

Making a clipping mask

This example shows how to create a clipping mask using the first path item in a group item. This is the
same effect as you get when you use the **Object > Clipping Mask > Make** command in the user interface.

```
-- Create a group of paths, then clip the group to the first path in the group
tell application "Adobe Illustrator"

    -- Create a group to contain the paths to be clipped
    set groupRef to make new group item in document 1

    -- Add some path items to the group
    make new rectangle at end of groupRef with properties ¬
        {bounds:{200, 350, 300, 250}, fill color:{cyan:100}, stroked:false}
    make new rectangle at end of groupRef with properties ¬
        {bounds:{300, 250, 400, 150}, fill color:{magenta:100}, stroked:false}
    make new rectangle at end of groupRef with properties ¬
        {bounds:{300, 350, 400, 250}, fill color:{yellow:100}, stroked:false}
    make new rectangle at end of groupRef with properties ¬
        {bounds:{200, 250, 300, 150}, fill color:{green:255}, stroked:false}

    -- Get a little fancy and create a rotated star at the center of the group
    set pathRef to make new star at beginning of groupRef with properties ¬
        {center point:{300, 250}, radius:25, inner radius:4, point count:4 ¬
            , fill color:{black:100}, opacity:40, stroked:false}
    set rotationMatrix to get rotation matrix angle 45
    transform pathRef using rotationMatrix about center

    -- Create the path that the group will be clipped with
    --    The clipping path must be the first (frontmost) path in the group
    make new star at beginning of groupRef with properties ¬
        {center point:{300, 250}, radius:80, inner radius:25, point count:4 ¬
            , stroked:false, filled:false}

    -- Now clip the group to the top path
    set clipped of groupRef to true
end tell
```

Illustrator preferences

Specifies the preferred options for AutoCAD, FreeHand, PDF, and Photoshop files.

Preference accessor guidelines

Preference accessor commands, such as `get boolean preference`, should be used only as a solution of last resort. For information on preference keys that can be accessed, see the `AIPreferenceKeys.h` header file in the Adobe Illustrator SDK.

NOTE: Preference keys other than those documented in `AIPreferenceKeys.h` are subject to change without notice and should not be used.

Illustrator preferences object properties

Property	Value type	What it is
`AutoCAD file options`	AutoCAD options	Read-only. Options to use when opening or placing an AutoCAD file.
`best type`	type class	Read-only. The best type for the object's value.
`class type`	type class	Read-only. The object's class.
`default type`	type class	Read-only. The default type for the object's value.
`FreeHand file options`	FreeHand options	Read-only. Options to use when opening or placing a FreeHand file.
`properties`	record	All properties of this object returned as a record.
`PDF file options`	PDF options	Read-only. Options to use when opening or placing a PDF file.
`Photoshop file options`	Photoshop options	Read-only. Options to use when opening or placing a Photoshop file.

Illustrator save options

Options that may be supplied when saving a document as an Illustrator file. All properties are optional.

See the save command for additional details.

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

Illustrator save options object properties

Property	Value type	What it is
`artboard range`	string	If save multiple artboards is true (which is valid only for Illustrator 13 or earlier), the document is considered for multi-asset extraction, which specifies an artboard range. An empty string extracts all artboards. Default: empty string

`compatibility`	Valid values:
	`Illustrator 8`
	`Illustrator 9`
	`Illustrator 10`
	`Illustrator 11`
	`Illustrator 12`
	`Illustrator 13`
	`Illustrator 14`
	`Illustrator 15`
	`Illustrator 16`
	`Japanese version 3`
		Specifies the version of the Illustrator file format to create. Default: `Illustrator 16`

`compressed`	`boolean`	If `true`, the saved file should be compressed. Only for Illustrator 10 or later. Default: `true`
`embed ICC profile`	`boolean`	If `true`, the document's ICC profile should be embedded in the saved file. Only for Illustrator 9 or later. Default: `false`.
`embed linked files`	`boolean`	If `true`, include linked image files in the saved document. Only for Illustrator 7 or later. Default: `false`.

`flatten output`	Valid values:
	`preserve paths`
	`preserve appearance`
		How should transparency be flattened for file formats before Illustrator 9 or later. Default: `preserve appearance`.

`font subset threshold`	real	Include a subset of fonts when less than this percentage of characters are used. Only for Illustrator 9 or later. Range: 0.0 to 100.0. Default: 100.0.

Property	Value type	What it is
PDF compatible	boolean	If `true`, the file should be saved as a PDF compatible file. Only for Illustrator 10 or later.
save multiple artboards	boolean	If `true`, all artboards or range of the artboards are saved. Valid for Illustrator 13 or earlier.

Valid commands

save

Save files in a folder

This handler processes a folder of Illustrator files, saving each with Illustrator 7 compatibility. Note that the `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destinationFolder is an alias to a folder where the files are to be saved

on SaveFilesAsIllustrator(fileList, filePath, destinationFolder)
    set destinationPath to destinationFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destinationPath & fileName & ".ai"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                save current document in file newFilePath as Illustrator with options
{flatten output:preserve appearance, compressed:true}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsIllustrator
```

image capture options

Options for image capture, used with the hide perspective grid command. All properties are optional.

ImageCaptureOptions properties

Property	Value type	What it is
`antialiasing`	`boolean`	If `true`, the image result is anti-aliased. Default: `false`
`matte`	`boolean`	If `true`, the artboard is matted with a color. Default: `false`
`matte color`	RGB color info	The color to use for the artboard matte. Default: `white`
`resolution`	`real`	The resolution of the captured image file in points-per-inch (PPI), in the range [72.0 ... 2400.0]. Default: 150
`transparency`	`boolean`	If `true`, the image result is transparent. Default: `false`

ink

Specifies the properties of the inks to be used in printing the document.

ink object properties

Property	Value type	What it is
name	Unicode text	The ink's name.
properties	ink properties	The ink information.

List inks in a document

```
-- Create a new CMYK document
-- Get the name of every ink in document 1
-- Display the list of ink names in a text frame
tell application "Adobe Illustrator"
    set inkNames to ""
    set theText to ""
    if not (exists document 1) then error "There is no available document."
    get the name of every item of inks of document 1
    repeat with theName in the result
        set inkNames to inkNames & theName & return
    end repeat
    set theText to inkNames
    set textRef to make new text frame in current document with properties
{position:{100, 500}}
    set contents of textRef to theText
end tell
```

ink properties

Information about ink use when printing a document with the <u>print</u> command.

ink properties object properties

Property	Value type	What it is
`angle`	`real`	The ink's screen angle in degrees. Range: -360 to 360
`custom color`	list of `real` numbers	The custom color.
`density`	`real`	The neutral density. Minimum: 0.0
`dot shape`	Unicode text	The dot shape name.
`frequency`	`real`	The ink's frequency. Range: 0.0 to 1000.0

`kind`	Valid values:
	`black ink`
	`custom ink`
	`cyan ink`
	`magenta ink`
	`yellow ink`
		The ink type.

`printing status`	Valid values:
	`convert ink`
	`disable ink`
	`enable ink`
		The ink printing status.

`trapping`	Valid values:
	`ignore opaque`
	`normal`
	`opaque`
	`transparent`
		The trapping type.

`trapping order`	`integer`	The order of trapping for the ink. Range: 1 to 4 for CMYK

insertion point

A location between characters, used to insert new text objects.

An insertion point is logically located between two characters in a text frame. Each insertion point is before the corresponding character in a text frame. Insertion point 1 is before character 1, etc.

The properties of an insertion point are the same as the character at the same position in the text frame. For example, the font for insertion point 2 of text frame 1 will be the same as the font for character 2 of text frame 1.

You can set the properties for an insertion point, but setting only the contents property has no affect on the text frame. The result of setting the contents of an insertion point to a string value is to insert the string in the text frame at the insertion point's location. Setting the contents to an empty string has no affect.

An insertion point is contained in an `InsertionPoints` collection. This is a [text] object in which `character offset` indicates the location of the insertion point and `length` is 0. This subclass does not define any additional properties.

insertion point object elements

Elements	Refer to by
character style	name, numeric index, range of elements, before/after another element, satisfying a test
character	numeric index, range of elements, before/after another element, satisfying a test
insertion point	numeric index, range of elements, before/after another element, satisfying a test
line	numeric index, range of elements, before/after another element, satisfying a test
paragraph style	name, numeric index, range of elements, before/after another element, satisfying a test
paragraph	numeric index, range of elements, before/after another element, satisfying a test
text	numeric index, range of elements, before/after another element, satisfying a test
word	numeric index, range of elements, before/after another element, satisfying a test

insertion point object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
container	reference	Read-only. The object's container.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.

Property	Value type	What it is
properties	record	All properties of this object returned as a record.
story	story	Read-only.The story that contains the insertion point.

insertion point object commands

<u>count</u>
<u>exists</u>

Working with insertion points

This example shows several ways of working with insertion points.

```
tell application "Adobe Illustrator"
    -- Set insertion point karat to beginning of a text frame
    set selection to insertion point 1 of text frame 1 of document 1
    -- Add a string to end of a text frame
    get insertion point -1 of text frame 1 of document 1
    make new word at (item 1 of the result) with properties {contents:"Some new text."}
    -- Since the default type of an insertion point is string, asking for
    -- a particular insertion point returns its contents. To get a reference
    -- to an insertion point you need to ask for a reference
    set insertionRef to ¬
        insertion point after word 3 of text frame 1 of document 1 as reference
    make new word at insertionRef with properties {contents:"more words"}
end tell
```

Add a word at the insertion point

```
--  Make a new document
--- Make a new text frame with contents "Wouldn't you rather be scripting?"
--  Change the size of the text frame
--  Get the insertion points of the last word of the text frame
--  Add a new word at the first insertion point of the result

tell application "Adobe Illustrator"
    activate
    make new document
    make new text frame in document 1 with properties {contents:"Wouldn't you rather be
scripting?", position:{100, 400}}
    set the size of the text of the result to 20
    delay 1
    get insertion points of the last word of text frame 1 of document 1
    make new word at (item 1 of the result) with properties {contents:"AppleScript"}
end tell
```

job options

The print job options when printing a document with the [print](#) command.

job options object properties

Property	Value type	What it is
`artboard range`	`string`	The artboard range to be printed if `print all artboards` is false. Default: 1-
`bitmap resolution`	`real`	The bitmap resolution. Minimum: 0.0. Default: 0.0
`collate`	`boolean`	If `true`, collate print pages are collated. Default: `false`
`copies`	`integer`	The number of copies to print. Minimum: 1. Default: 1

`designation`	Valid values:
	   `all layers`
	   `visible layers`
	   `visible printable layers`
		The layers/objects to be printed. Default: `visible printable layers`

`file path`	file specification	The file to which to print.
`name`	Unicode text	The print job name.
`print all artboards`	`boolean`	Indicates whether to print all artboards. Default: `true`

`print area`	Valid values:
	   `artboard bounds`
	   `artwork bounds`
		The printing bounds. Default: `artboard bounds`

`print as bitmap`	`boolean`	If `true`, the job is printed as a bitmap image. Default: `false`
`reverse pages`	`boolean`	If `true`, the pages are printed in reverse order. Default: `false`

Print job options

```
-- Makes new document containing 3 layers - 1
-- non printable, 1 non visible and 1 visible and printable
-- a text frame is added to each layer
-- Print all layers
-- Print only visible layers
-- Print only visible and printable layer
tell application "Adobe Illustrator"
    activate
    make new document
    set the name of current layer of document 1 to "VPL"
    make new text frame in document 1 with properties {contents:"Visible and Printable",
position:{200, 600}}
    make new layer in document 1 with properties {name:"VnPL", printable:false}
    make new text frame in layer "VnPL" of document 1 with properties {contents:"Visible
and Non-Printable", position:{200, 500}}
    make new layer in document 1 with properties {name:"nVPL"}
    make new text frame in layer "nVPL" of document 1 with properties
{contents:"Non-Visible", position:{200, 400}}
    set visible of layer "nVPL" of document 1 to false
    set printOptions to {class:print options, job settings:{class:job options,
designation:all layers, reverse pages:true}}
    print document 1 options printOptions
    set printOptions to {class:print options, job settings:{class:job options,
designation:visible layers, reverse pages:true}}
    print document 1 options printOptions
    set jobOptions to {class:job options, designation:visible printable layers, reverse
pages:true}
    set printOptions to {class:print options, job settings:jobOptions}
    print document 1 options printOptions
end tell
```

JPEG export options

Options that can be supplied when exporting a document as a JPEG file. See the <u>export</u> command in the command reference for additional details.

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

JPEG export options object properties

Property	Value type	What it is
`antialiasing`	boolean	If `true`, the resulting image should be anti-aliased. Default: `true`
`artboard clipping`	boolean	If `true`, the resulting image should be clipped to the artboard. Default: `false`
`blur`	real	The amount of blurring to apply to the resulting image. Range: 0.0 to 2.0. Default: 0.0
`horizontal scaling`	real	The percent horizontal scaling factor to apply to the resulting image. Range: 0.0 to 100.0. Default: 100.0
`matte`	boolean	If `true`, the artboard should be matted with a color. Default: `true`
`matte color`	<u>RGB color info</u>	The color to use when matting the artboard. Default: `white`
`optimization`	boolean	If `true`, the resulting image should be optimized for web viewing. Default: `true`
`quality`	integer	The quality of the resulting image. Range: 0 to 100. Default: 30
`saving as HTML`	boolean	If `true`, the resulting image should be saved with an accompanying HTML file. Default: `false`
`vertical scaling`	real	The percent vertical scaling factor to apply to the resulting image. Range: 0.0 to 776.19. Default: 100.0

Export to JPEG

This handler processes all Illustrator files in a specific folder, exporting each file as a medium-quality JPEG image. Note that the `class` property is specified in the record to ensure that Illustrator can determine the export option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destinationFolder is an alias to a folder where the files are to be saved

on ExportFilesAsJPEGMedium(fileList, filePath, destinationFolder)
    set destinationPath to destinationFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destinationPath & fileName & ".jpg"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                export current document to file newFilePath as JPEG with options ¬
                    {class:JPEG export options ¬
                        , quality:60 ¬
                        , blur:0.5 ¬
                        , horizontal scaling:50.0 ¬
                        , vertical scaling:50 ¬
                        , matte:false}
                close current document saving no
            end tell
        end repeat
    end if
end ExportFilesAsJPEGMedium
```

Lab color info

A color specification in the CIE Lab color space, used where a `color info` object is required.

Lab color info properties

This class inherits all properties from the `color info` class.

Property	Value type	What it is
a	real	The a (red-green) color value. Range -128.0–128.0. Default: 0.0
b	real	The b (yellow-blue) color value. Range -128.0–128.0. Default: 0.0
l	real	The l (lightness) color value. Range -128.0–128.0. Default: 0.0

layer, layers

A layer or list of layers. Layers may contain nested layers, which are called sublayers in the user interface.

The `layer` object contains all the page items in the specific layer as elements. Your script can access page items as elements of either the `layer` object or as elements of the `document` object. When accessing page items as elements of a layer, only objects in that layer can be accessed. To access page items throughout the entire document, be sure to refer to them as elements of the document.

layer object elements

Element	Refer to by
compound path item	name, index, before/after, range, test
graph item	name, index, before/after, range, test
group item	name, index, before/after, range, test
layer	name, index, before/after, range, test
legacy text item	name, index, before/after, range, test
mesh item	name, index, before/after, range, test
non native item	name, index, before/after, range, test
page item	name, index, before/after, range, test
path item	name, index, before/after, range, test
placed item	name, index, before/after, range, test
plugin item	name, index, before/after, range, test
raster item	name, index, before/after, range, test
symbol item	name, index, before/after, range, test
text frame	name, index, before/after, range, test

layer object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the layer object's value. Always returns `reference`.

blend mode	Valid values:
	  `color blend`
	  `color burn`
	  `color dodge`
	  `darken`
	  `difference`
	  `exclusion`
	  `hard light`
	  `hue`
	  `lighten`
	  `luminosity`
	  `multiply`
	  `normal`
	  `overlay`
	  `saturation blend`
	  `screen`
	  `soft light`
		The mode used when compositing an object. An object is considered composited when its opacity is set to less than 100.0 (100%).

class	type class	Read-only. The layer object's class, which is `layer`.
color	[RGB color info](#)	The layer's selection mark color.
container	object reference	Read-only. A reference to the document that contains this layer.
default type	type class	Read-only. The default type for the layer object's value. Always returns `reference`.
dim placed images	`boolean`	If `true`, placed images are to be rendered as dimmed in this layer.
has selected artwork	`boolean`	If `true`, one or more objects in this layer selected are selected; setting this property to `false` deselects all objects in the layer.
index	`integer`	Read-only. The position of this layer in the current stacking order of layers in this document, where `layer 1` is always the topmost layer in the stacking order.
isolated	`boolean`	If `true`, this object is isolated

knockout	Valid values:
	  `unknown`
	  `disabled`
	  `enabled`
	  `inherited`
		Is this object used to create a knockout.

locked	`boolean`	If `true`, the `layer` is editable.
name	Unicode text	The name of this layer.

Property	Value type	What it is
opacity	`real`	The opacity of this layer, where 100.0 is completely opaque and 0.0 is completely transparent.
preview	`boolean`	If `true`, this layer should be displayed using preview mode.
printable	`boolean`	If `true`, this layer should be printed when printing the document.
properties	`record`	All properties of this object returned as a record.
sliced	`boolean`	If `true`, slices should be preserved. Default: `false`
visible	`boolean`	If `true`, this layer is visible.

layer object commands

count
delete
duplicate
exists
make
move

Move layers

```
-- Move the 2nd layer to the top of the stacking order
tell application "Adobe Illustrator"
    if (count layers of current document) > 1 then
        move layer 2 of document 1 to before layer 1 of document 1
    end if
end tell
```

Create a layer

Commands that deal with changes to an object's reference, including the creation of new objects with the `make` command, return a reference to the new or modified object in their result. This example stores the reference returned for a newly created layer and then creates a new path item in the layer using the reference.

```
-- Make a new layer at the top of the layer stack
-- then create a new path in the layer
tell application "Adobe Illustrator"
    set layerRef to make layer at document 1 with properties {name:"Our Layer"}
    make new rectangle at beginning of layerRef
end tell
```

Delete layers

This example demonstrates the power of constructing simple tests (with the `whose` clause) to selectively delete layers in a document based on their names. In this case, the script deletes all layers in the current document that have names starting with the word "Temporary."

```
-- Delete layers that have a name which begin with a particular string
set partialName to "Temp"
tell application "Adobe Illustrator"
    delete (every layer of document 1 whose name starts with partialName)
end tell
```

legacy text item, legacy text items

A text item from a document in a pre-CS version of Illustrator (version 10 or earlier), or a list of such items., which are uneditable until converted. To convert legacy text, see [convert](#).

You can view, move, and print legacy text, but you cannot edit it. Legacy text has an `"x"` through its bounding box when selected.

legacy text item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
converted	boolean	When `true`, the item has been updated to the current text format (a `text frame`). Read-only.
properties	record	All properties of this object returned as a record.

legacy text item object commands

[convert](#)

line

A line or lines of text in a text frame. A document's text can be accessed using the `character`, `insertion point`, `word`, `line`, `paragraph`, and `text` classes.

Lines of text cannot be created. When the `contents` property of a text frame is modified, Illustrator will create text lines as it reflows the text within the text frame.

line object elements

Elements	Refer to by
character style	name, numeric index, range of elements, before/after another element, satisfying a test
character	numeric index, range of elements, before/after another element, satisfying a test
insertion point	numeric index, range of elements, before/after another element, satisfying a test
line	numeric index, range of elements, before/after another element, satisfying a test
paragraph style	name, numeric index, range of elements, before/after another element, satisfying a test
paragraph	numeric index, range of elements, before/after another element, satisfying a test
text	numeric index, range of elements, before/after another element, satisfying a test
word	numeric index, range of elements, before/after another element, satisfying a test

line object properties

Property	Value type	What it is
aki left	real	The amount of inter-glyph space added to the left side of each glyph in Japanese text (in thousandths of an em).
aki right	real	The amount of inter-glyph spacing added to the right side of each glyph in Japanese text (in thousandths of an em).
alignment	Valid values: bottom center icf bottom icf top roman baseline top	The character alignment type.

Property	Value type	What it is

alternate glyphs	Valid values:
	   `default`
	   `expert`
	   `full width`
	   `half width`
	   `jis78`
	   `jis83`
	   `proportional width`
	   `quarter width`
	   `third width`
	   `traditional`
	   `jis90`
	   `jis04`
		Specifies the type of alternate glyphs.

alternate ligature	`boolean`	If `true`, use the alternate ligature.
auto leading	`boolean`	If `true`, use automatic leading.

baseline direction	Valid values:
	   `standard`
	   `Tate Chu Yoko`
	   `vertical rotated`
		The Japanese text baseline direction.

baseline position	Valid values:
	   `normal`
	   `subscript`
	   `superscript`
		The baseline position of text.

baseline shift	`real`	The amount of shift (in points) of the text baseline.
best type	type class	Read-only. The best type for the object's value.

capitalization	Valid values:
	   `all caps`
	   `all small caps`
	   `normal`
	   `small caps`
		The case of the text.

character offset	`integer`	Offset of the first character.
class	type class	Read-only. The object's class.
connection forms	`boolean`	If `true`, use the OpenType connection forms.
container	`reference`	Read-only. The object's container.
contents	Unicode text	The text content.
contextual ligature	`boolean`	If `true`, use the contextual ligature.
default type	type class	Read-only. The default type for the object's value.
discretionary ligature	`boolean`	If `true`, use the discretionary ligature.

Property	Value type	What it is

figure style	Valid values:
	    default
	    proportional
	    proportional oldstyle
	    tabular
	    tabular oldstyle
		Specifies which figure style to use in an OpenType font.

fill color	color info	The color of the text fill.
fractions	boolean	If `true`, use the OpenType fractions.
horizontal scale	real	The character horizontal scaling factor expressed as a percentage (100 = 100%).
index	integer	Read-only. The index of this instance of the object
italics	boolean	If `true`, the Japanese OpenType support supports the italic style.
kerning	integer	Controls the spacing between two characters, in thousandths of an em.

kerning method	Valid values:
	    auto
	    none
	    optical
		The automatic kerning method to use.

Property	Value type	What it is

`language`	Valid values:
	  Bokmal Norwegian
	  Brazillian Portuguese
	  Bulgarian
	  Canadian French
	  Catalan
	  Chinese
	  Czech
	  Danish
	  Dutch
	  Dutch 2005 Reform
	  English
	  Finnish
	  German 2006 Reform
	  Greek
	  Hungarian
	  Icelandic
	  Italian
	  Japanese
	  Nynorsk Norwegian
	  old German
	  Polish
	  Romanian
	  Russian
	  Spanish
	  Serbian
	  standard French
	  standard German
	  standard Portuguese
	  Swedish
	  Swiss German
	  Swiss German 2006 Reform
	  Turkish
	  UK English
	  Ukranian
		The language.

`leading`	real	The amount of space between two lines of text, in points.
`length`	integer	The length in characters. Minimum: 0
`ligature`	boolean	If `true`, use the ligature.
`no break`	boolean	Whether break is allowed.

`OpenType position`	Valid values:
	  default
	  denominator
	  numerator
	  subscript
	  superscript
		The OpenType baseline position.

`ordinals`	boolean	If `true`, use the OpenType ordinals.
`ornaments`	boolean	If `true`, use the OpenType ornaments.
`overprint fill`	boolean	If `true`, overprint the fill of the text.

Property	Value type	What it is
overprint stroke	boolean	If true, the stroke of the text may be overprinted.
properties	record	All properties of this object returned as a record.
proportional metrics	boolean	If true, the proportional metrics in Japanese OpenType may be used.
rotation	real	The character rotation angle.
selection	list of text	Read-only. The selected text.
size	real	Font size in points.
story	story	Read-only. The story that contains the line.
strike through	boolean	If true, characters use strike-through style.
stroke color	color info	The color of the text stroke.
stroke weight	real	line width of stroke.
stylistic alternates	boolean	If true, use the OpenType stylistic alternates.
swash	boolean	If true, use the OpenType swash.
TCY horizontal	integer	The Tate-Chu-Yoko horizontal adjustment in points.
TCY vertical	integer	The Tate-Chu-Yoko vertical adjustment in points.
text font	text	The text font.
titling	boolean	If true, use the OpenType titling alternates.
tracking	integer	The tracking or range kerning amount in thousandths of an em.
Tsume	real	The percentage of space reduction around a Japanese character.
underline	boolean	If true, characters use underline style.
vertical scale	real	Character vertical scaling factor.
warichu characters after break	integer	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu characters before break	integer	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.

Property	Value type	What it is
`warichu enabled`	`boolean`	If `true`, Wari-Chu is enabled.
`warichu gap`	`integer`	The Wari-Chu line gap.

`warichu justification`	Valid values:
	`auto justify`
	`center`
	`full justify last line center`
	`full justify`
	`full justify last line left`
	`full justify last line right`
	`left`
	`right`
		The Wari-Chu justification.

`warichu lines`	`integer`	The number of Wari-Chu (multiple text lines fit into a space meant for one) lines.
`warichu scale`	`real`	The Wari-Chu scale.

line object commands

[apply character style](#)
[apply paragraph style](#)
[change case](#)
[count](#)
[delete](#)
[deselect](#)
[duplicate](#)
[exists](#)
[make](#)
[move](#)
[select](#)

Finding lines of text

Lines of text can be located with matching characteristics using the `whose` clause, as this script demonstrates.

```
-- Color red all lines of text containing more than 10 characters
tell application "Adobe Illustrator"
    if (count text frames in document 1) > 0 then
        set textItemCount to count text frames in document 1
        repeat with i from 1 to textItemCount
            set (fill color of every line of text frame i of document 1 ¬
                whose length > 10) to {red:255.0}
        end repeat
    end if
end tell
```

matrix

A transformation matrix specification, used to transform the geometry of objects.

Matrices are used in conjunction with the `transform` command and as a property of a number of objects. You can generate an original matrix using the `get identity matrix`, `get translation matrix`, `get scale matrix`, or `get rotation matrix` commands.

A `matrix` is a record containing the matrix values, not a reference to a matrix object. The matrix commands listed above operate on the values of a matrix record. If a command modifies a matrix, a modified matrix record is returned as the result of the command. The original matrix record passed to the command is not modified.

matrix object properties

Property	Value type	What it is
mvalue_a	real	Matrix property `a`.
mvalue_b	real	Matrix property `b`.
mvalue_c	real	Matrix property `c`.
mvalue_d	real	Matrix property `d`.
mvalue_tx	real	Matrix property `tx`.
mvalue_ty	real	Matrix property `ty`.

matrix object commands

[concatenate matrix](#)
[concatenate rotation matrix](#)
[concatenate scale matrix](#)
[concatenate translation matrix](#)
[equal matrices](#)
[get identity matrix](#)
[get rotation matrix](#)
[get scale matrix](#)
[get translation matrix](#)
[invert matrix](#)
[singular matrix](#)

Getting a matrix for scale transformation

A matrix can be generated to effect a scale transformation using the `get scale matrix` command.

```
-- Scale all art in a document to 50% vertical size
tell application "Adobe Illustrator"
    if (count page items in document 1) > 0 then
        set scaleMatrix to get scale matrix horizontal scale 100.0 vertical scale 50.0
        transform every page item in document 1 using scaleMatrix
    end if
end tell
```

Applying multiple transformations

To apply multiple transformations to objects, it is more efficient to use the matrix suite than to apply the transformations one at a time. The following script demonstrates how to combine multiple matrices.

```
-- Scale, rotate, and translate all art in a document
tell application "Adobe Illustrator"
    if (count page items in document 1) > 0 then
        set matrixDef to ¬
            get scale matrix horizontal scale 100.0 vertical scale 50.0
        set matrixDef to ¬
            concatenate rotation matrix matrixDef angle -45.0
        set matrixDef to ¬
            concatenate translation matrix matrixDef delta x 50.0 delta y -50.0
        transform every page item in document 1 using matrixDef
    end if
end tell
```

mesh item, mesh items

A gradient mesh art item or list of gradient mesh art items. Scripts cannot create new mesh items, but can be duplicate, copy and paste them.

mesh item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
properties	record	All properties of this object returned as a record.

mesh item object commands

count
delete
duplicate
exists
move
rotate
scale
transform
translate

no color info

Represents the "none" color. Assigning a reference to a `no color` object to a document's default fill or stroke color, or those of an art item, is equivalent to setting their `filled` or `stroked` property to `false`.

This class inherits all properties from the [color info](#) class.

Setting color to none

```
-- Make a new document
-- Make two overlapping rectangles with different fill colors
-- Set the fill color of the top rectangle to no color
tell application "Adobe Illustrator"
    activate
    make new document with properties {color space:RGB}
    make new rectangle in document 1 with properties {position:{200, 500}, width:300,
height:100}
    set the fill color of the result to {class:RGB color info, red:255, green:0, blue:0}
    make new rectangle in document 1 with properties {position:{150, 550}, width:200,
height:100}
    set the fill color of the result to {class:RGB color info, red:0, green:255, blue:0}
    delay 1
    set the fill color of path item 1 of document 1 to {class:no color info}
end tell
```

non native item, non native items

A non-native artwork item or a list of those items.

non native item object properties

These classes inherit all properties from the `page item` class.

Property	Value type	What it is
properties	record	All properties of this object returned as a record.

non native item object commands

count
delete
duplicate
exists
move
rotate
scale
transform
translate

open options

Specifies options that can be supplied when opening a file.

open options object properties

Property	Value type	What it is

`as`	Valid values:
	`Illustrator artwork`
	`swatches library`
	`brushes library`
	`graphic styles library`
	`symbols library`
		Open as an Illustrator library of the given type. Default: `Illustrator artwork`

`convert crop area to artboard`	`boolean`	If `true`, converts crop areas to artboards when opening a legacy document in Illustrator CS4 or later. When false, crop areas are discarded. Default: `true`.
`convert tiles to artboard`	`boolean`	If `true`, converts print tiles to artboards when opening a legacy document in Illustrator CS4 or later. Default: `false`.
`create artboard with artwork bounding box`	`boolean`	If `true`, creates an artboard with the dimensions of the bounding box of the artwork when opening a legacy document in Illustrator CS4 or later. Default: `false`.
`preserve legacy artboard`	`boolean`	If `true`, preserves legacy artboards when opening a legacy document in Illustrator CS4 or later. Default: `true`.
`update legacy gradient mesh`	`boolean`	If `true`, preserves the spot colors in the gradient mesh objects for legacy documents (pre-Illustrator CS4). Default: `true`
`update legacy text`	`boolean`	Read-only. If `true`, update all legacy text objects for documents saved with Illustrator version 10 or earlier. Default: `false`

Open a file with automatic update of legacy text

```
-- This function opens a file passed to it, any
-- legacy text is automatically updated, fileToOpen
-- is set by the framework this fragment is tested in
on openLegacyFile(fileToOpen)
    tell application "Adobe Illustrator"
        activate
        open POSIX file fileToOpen as alias with options {update legacy text:true}
    end tell
end openLegacyFile
```

page item, page items

Any art item or list of art items. Every art item and group in a document is a page item. You may refer to a page item as an element of a document, layer, or group item.

The `page item` class gives you complete access to every art item contained in an Illustrator document. The `page item` class is the superclass of all artwork objects in a document. The `compound path item`, `group item`, `mesh item`, `non native item`, `path item`, `placed item`, `plugin item`, `raster item`, and `text frame` classes each inherit a set of properties from the `page item` class.

You cannot create a `page item` directly, you must create one of the specific `page item` subclasses, such as `path item`.

page item object elements

Element	Refer to by
`tag`	name, index, before/after, range, test

page item object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `page item` object's value. Always returns `reference`.

blend mode	Valid values:
	`color blend`
	`color burn`
	`color dodge`
	`darken`
	`difference`
	`exclusion`
	`hard light`
	`hue`
	`lighten`
	`luminosity`
	`multiply`
	`normal`
	`overlay`
	`saturation blend`
	`screen`
	`soft light`
		The mode to use when compositing this object. An object is considered composited when its opacity is set to less than 100.0 (100%).

class	type class	Read-only. The page item object's class, which can be any one of the specific classes that are children of the page item class, including `compound path item`, `group item`, `mesh item`, `non native item`, `path item`, `placed item`, `plugin item`, `raster item`, and `text frame`.
container	object reference	Read-only. A reference to the layer that contains this `page item`.

Property	Value type	What it is
control bounds	`list`	Read-only. The bounds of the object including stroke width and controls.
default type	type class	Read-only. The default type for the page item object's value. Always returns `reference`.
editable	`boolean`	Read-only. If `true`, this `page item` is editable.
geometric bounds	`list`	Read-only. The object's bounds excluding the stroke width.
height	`real`	The height of the `page item`, calculated from the geometric bounds.
hidden	`boolean`	If `true`, this `page item` is hidden.
index	`integer`	Read-only. The position of this `page item` in the current stacking order of the containing layer, where `page item` 1 is always topmost.
isolated	`boolean`	If `true`, this object is isolated.

knockout	Valid values:
	`unknown`
	`disabled`
	`enabled`
	`inherited`
		Is this object used to create a knockout.

layer	object reference	Read-only. The layer to which this `page item` belongs.
locked	`boolean`	If `true`, this `page item` is locked.
name	Unicode text	The name of this `page item`.
note	Unicode text	The note assigned to this item.
opacity	`real`	The opacity of this object, where 100.0 is completely opaque and 0.0 is completely transparent.
pixel aligned	`boolean`	True if the item is aligned to the pixel grid.
position	`fixed point`	The position (in points) of the top left corner of the item in the format {x, y}. Does not include stroke weight.
properties	`record`	All properties of this object returned as a record.
selected	`boolean`	If `true`, this object is selected.
sliced	`boolean`	If `true`, preserve slices.
URL	Unicode text	The value of the Adobe URL tag assigned to this `page item`.
visibility variable	anything	The visibility variable to which this `page item` path is bound.
visible bounds	[rectangle](#)	Read-only. The object's visible bounds, including stroke width of any objects in the illustration.

Property	Value type	What it is
width	real	The width of the `page item`, calculated from the geometric bounds.
wrap inside	boolean	If `true`, the text frame object should be wrapped inside this object.
wrap offset	number (double)	The offset to use when wrapping text around this object.
wrapped	boolean	If `true`, wrap text frame objects around this object (text frame must be above the object).

page item object commands

[bring in perspective](#)
[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[move](#)
[rotate](#)
[scale](#)
[transform](#)
[translate](#)

Move a page item

The stacking order of existing page items in a layer can be manipulated using the `move` command. This example demonstrates how to move a page item to the top of the stacking order (index position 1) in a layer.

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

page marks options

Specifies the page marks options when printing a document with the <u>print</u> command.

page marks options object properties

Property	Value type	What it is
bleed offset	list	The bleed offset rectangle.
color bars	boolean	If `true`, color bar printing is enabled. Default: `false`
marks offset	list	The page marks offset rectangle.
page info marks	boolean	If `true`, page info marks printing is enabled. Default: `false`
page marks style	Valid values: Japanese Roman	The page marks style. Default: `Roman`
registration marks	boolean	If `true`, the registration marks are printed. Default: `false`
trim marks	boolean	If `true`, printing of trim marks is enabled. Default: `false`
trim marks weight	real	Stroke weight of trim marks. Minimum: 0.0. Default: 0.125

Print page marks

```
-- Make sure a document is available
-- Create a page mark options object
-- Print the document with the page mark options
tell application "Adobe Illustrator"
    activate
    if not (exists document 1) then error "There is no available document."
    set pageMarkOptions to {class:page marks options, color bars:true, page info
marks:true, registration marks:true, trim marks:true}
    set printOptions to {class:print options, page marks settings:pageMarkOptions}
    print document 1 options printOptions
end tell
```

paper

This class contains information about the paper to be used when printing a document with the <u>print</u> command.

paper object properties

Property	Value type	What it is
name	Unicode text	The paper name.
properties	<u>paper properties</u>	The paper information.

paper options

Information about the paper options when printing a document with the [print](#) command.

paper options object properties

Property	Value type	What it is
height	real	Custom paper's height in points. Minimum 0.0. Default: 0.0
name	Unicode text	The paper's name.
offset	real	Custom paper's offset in points. Minimum 0.0. Default: 0.0
transverse	boolean	If `true`, transverse the artwork (rotate 90 degrees) on the custom paper. Default: `false`
width	real	Custom paper's width. Minimum 0.0. Default: 0.0

paper properties

Information about the paper.

paper properties object properties

Property	Value type	What it is
custom paper	boolean	If `true`, it is a custom paper.
height	real	The paper's height in points.
imageable area	list	The imageable area, a rectangle.
width	real	The paper's width in points.

Paper size

```
-- Make new document
-- Make a rectangle and apply a graphic style
-- Get the printer name of the first printer
-- Get the paper name of the first paper of the first printer
-- Print the document to the printer using the paper name as its paper option
tell application "Adobe Illustrator"
    activate
    make new document
    make new rectangle in document 1 with properties {position:{200, 600}, height:400,
width:100}
    apply graphic style 2 of document 1 to path item 1 of document 1
    if printers is not {} then
        set printerName to (name of item 1 of printers) as string
        set printerRef to item 1 of printers
        if paper sizes of properties of printerRef is not {} then
            set paperName to name of item 1 of paper sizes of properties of (get
properties of item 1 of printers)
            set paperOptions to {class:paper options, name:paperName}
            set printOptions to {class:print options, printer name:printerName, paper
settings:paperOptions}
            print document 1 options printOptions
        end if
    end if
end tell
```

paragraph, paragraphs

A paragraph or list of paragraphs of text in the contents of a text art item. A document's text can be accessed using the `character`, `insertion point`, `word`, `line`, `paragraph` and `text` classes. All text is contained within text frames.

The `paragraph` class has additional properties that other related classes do not share, including properties for margins, tab stop settings, hyphenation, and word/letter spacing.

paragraph object elements

Elements	Refer to by
`character style`	name, numeric index, range of elements, before/after another element, satisfying a test
`character`	numeric index, range of elements, before/after another element, satisfying a test
`insertion point`	numeric index, range of elements, before/after another element, satisfying a test
`line`	numeric index, range of elements, before/after another element, satisfying a test
`paragraph style`	name, numeric index, range of elements, before/after another element, satisfying a test
`paragraph`	numeric index, range of elements, before/after another element, satisfying a test
`text`	numeric index, range of elements, before/after another element, satisfying a test
`word`	numeric index, range of elements, before/after another element, satisfying a test

paragraph object properties

Property	Value type	What it is
`aki left`	`real`	The amount of extra space (aki) added to the left side of each glyph in Japanese text (in thousandths of an em).
`aki right`	`real`	The amount of extra space (aki) added to the right side of each glyph in Japanese text (in thousandths of an em).

`alignment`	Valid values:
	`bottom`
	`center`
	`icf bottom`
	`icf top`
	`roman baseline`
	`top`
		The character alignment type.

Property	Value type	What it is

alternate glyphs	Valid values:
	    default
	    expert
	    full width
	    half width
	    jis78
	    jis83
	    proportional width
	    quarter width
	    third width
	    traditional
	    jis90
	    jis04
		The type of alternate glyphs.

auto leading	boolean	If true, automatic leading is used.
auto leading amount	real	The auto leading amount, as a percentage.
auto TCY	integer	The automatic Tate-Chu-Yoko amount.

baseline direction	Valid values:
	    standard
	    Tate Chu Yoko
	    vertical rotated
		The Japanese text baseline direction.

baseline position	Valid values:
	    normal
	    subscript
	    superscript
		The baseline position of text.

baseline shift	real	The amount of shift (in points) of the text baseline.
best type	type class	Read-only. The best type for the object's value.
BunriKinshi	boolean	If true, BunriKinshi is enabled.

Burasagari type	Valid values:
	    forced
	    none
	    standard
		The Burasagari type which specifies whether punctuation is allowed to fall outside of the paragraph bounding box (not available when Kinsoku Shori is set to None).

capitalization	Valid values:
	    all caps
	    all small caps
	    normal
	    small caps
		The case of the text.

character offset	integer	Offset of the first character.
class	type class	Read-only. The object's class.
connection forms	boolean	If true, use the OpenType connection forms.

Property	Value type	What it is
`container`	`reference`	Read-only. The object's container.
`contents`	Unicode text	The text content.
`contextual ligature`	`boolean`	If `true`, use the contextual ligature.
`default type`	type class	Read-only. The default type for the object's value.
`desired glyph scaling`	`real`	Desired glyph scaling, expressed as a percentage of the default character width. Range: 50.0 to 200.0; at 100.0, the width of characters is not changed.
`desired letter spacing`	`real`	Desired letter spacing, expressed as a percentage of the default kerning or tracking Range: -100.0 to 500.0; at 0, no space is added between letters; at 100.0, an entire space width is added between letters.
`desired word spacing`	`real`	Desired word spacing, expressed as a percentage of the default space for the font. Range: 0.0 to 1000.0; at 100.00, no space is added between words.
`discretionary ligature`	`boolean`	If `true`, use the discretionary ligature.
`every line composer`	`boolean`	If `true`, the Every-line Composer is enabled. If `false`, the Single-line Composer is enabled.

`figure style`	Valid values:
	  `default`
	  `proportional`
	  `proportional oldstyle`
	  `tabular`
	  `tabular oldstyle`
		The number style for OpenType font.

`fill color`	[color info](#)	The color of the text fill.
`first line indent`	`real`	First line left indent expressed in points.
`fractions`	`boolean`	If `true`, uses OpenType fractions.
`horizontal scale`	`real`	The character horizontal scaling factor expressed as a percentage (100 = 100%).
`hyphenate capitalized words`	`boolean`	If `true`, hyphenation is enabled for capitalized words.

Property	Value type	What it is
hyphenation	boolean	If `true`, hyphenation is enabled for the paragraph.
hyphenation preference	real	Hyphenation preference scale for better spacing (0) or fewer hyphens (1). Range: 0.0 to 1.0
hyphenation zone	real	The distance (in points) from the right edge of the paragraph that marks the part of the line where hyphenation is not allowed. 0 allows all hyphenation. Valid only when [every line composer](#) is `false`.
index	integer	The index of this instance of the object.
italics	boolean	If `true`, the Japanese OpenType support supports the italic style.
justification	Valid values: `center` `full justify last line center` `full justify last line full` `full justify last` `line left` `full justify last line right` `left` `right`	Paragraph justification.
kerning	integer	Controls the spacing between two characters, in thousandths of an em.
kerning method	Valid values: `auto` `none` `optical`	The automatic kerning method to use.
Kinsoku	Unicode text	The name of a Kinsoku Shori set (a set of characters which cannot be used to begin or end a line of Japanese text).
Kinsoku order	Valid values: `push in` `push out first` `push out only`	The preferred Kinsoku order.
KurikaeshiMojiShori	boolean	If `true`, Kurikaeshi Moji Shori is enabled (controls how repeated characters are handled in Japanese text).

Property	Value type	What it is

language	Valid values:
	  Bokmal Norwegian
	  Brazillian Portuguese
	  Bulgarian
	  Canadian French
	  Catalan
	  Chinese
	  Czech
	  Danish
	  Dutch
	  Dutch 2005 Reform
	  English
	  Finnish
	  German 2006 Reform
	  Greek
	  Hungarian
	  Icelandic
	  Italian
	  Japanese
	  Nynorsk Norwegian
	  old German
	  Polish
	  Romanian
	  Russian
	  Spanish
	  Serbian
	  standard French
	  standard German
	  standard Portuguese
	  Swedish
	  Swiss German
	  Swiss German 2006 Reform
	  Turkish
	  UK English
	  Ukranian
		The language.

leading	real	Specifies the amount of space between two lines of text (in points).

leading type	Valid values:
	  Japanese
	  Roman
		Auto leading type.

left indent	real	Left indent of margin expressed in points.
length	integer	The number of characters in the paragraph. Minimum: 0
ligature	boolean	If `true`, the ligature should be used.
maximum consecutive hyphens	integer	Maximum number of consecutive hyphenated lines.

Property	Value type	What it is
`maximum glyph scaling`	`real`	Maximum glyph scaling, expressed as a percentage of the default character width. Range: 50.0 to 200.0; at 100.0, the width of characters is not changed. **NOTE:** Valid only for justified paragraphs.
`maximum letter spacing`	`real`	Maximum letter spacing, expressed as a percentage of the default kerning or tracking Range: -100.0 to 500.0; at 0, no space is added between letters; at 100.0, an entire space width is added between letters. **NOTE:** Valid only for justified paragraphs.
`maximum word spacing`	`real`	Maximum word spacing, expressed as a percentage of the default space for the font. Range: 0.0 to 1000.0; at 100.00, no space is added between words. **NOTE:** Valid only for justified paragraphs.
`minimum after hyphen`	`integer`	Minimum number of characters after a hyphen.
`minimum before hyphen`	`integer`	Minimum number of characters before a hyphen.
`minimum glyph scaling`	`real`	Minimum glyph scaling, expressed as a percentage of the default character width. Range: 50.0 to 200.0; at 100.0, the width of characters is not changed. **NOTE:** Valid only for justified paragraphs.
`minimum hyphenated word size`	`integer`	Minimum number of characters for a word to be hyphenated.

Property	Value type	What it is
minimum letter spacing	real	Minimum letter spacing, expressed as a percentage of the default kerning or tracking Range: -100.0 to 500.0; at 0, no space is added between letters; at 100.0, an entire space width is added between letters. **NOTE:** Valid only for justified paragraphs.
minimum word spacing	real	Minimum word spacing, expressed as a percentage of the default space for the font. Range: 0.0 to 1000.0; at 100.00, no space is added between words **NOTE:** Valid only for justified paragraphs.
Mojikumi	Unicode text	The name of a predefined Mojikumi set for Japanese text composition.
no break	boolean	If `true`, a break is allowed.
ordinals	boolean	If `true`, use the OpenType ordinals.
ornaments	boolean	If `true`, use the OpenType ornaments.
overprint fill	boolean	If `true`, overprint the fill of the text.
overprint stroke	boolean	If `true`, the stroke of the text may be overprinted.

OpenType position	Valid values:
	    default
	    denominator
	    numerator
	    subscript
	    superscript
		The OpenType baseline position.

properties	record	All properties of this object returned as a record.
proportional metrics	boolean	If `true`, the proportional metrics in Japanese OpenType may be used.
right indent	real	Right indent of margin expressed in points.
roman hanging	boolean	If `true`, Roman hanging punctuation is enabled.
rotation	real	The character rotation angle.

Property	Value type	What it is
`selection`	list of <u>text</u>	The selected text.

`single word justification`	Valid values:
	  `center`
	  `full justify last line center`
	  `full justify`
	  `full justify last line left`
	  `full justify last line right`
	  `left`
	  `right`
		Justification type for a single word.

`size`	`real`	Font size in points.
`space after`	`real`	Spacing after paragraph in points.
`space before`	`real`	Spacing before paragraph in points.
`story`	`story`	The story in the paragraph.
`strike through`	`boolean`	If `true`, characters use strike-through style.
`stroke color`	<u>color info</u>	The color of the text stroke.
`stroke weight`	`real`	line width of stroke.
`stylistic alternates`	`boolean`	If `true`, use the OpenType stylistic alternates.
`swash`	`boolean`	If `true`, use the OpenType swash.
`tab stops`	list of `tab stop info`	Tab stop settings.
`TCY horizontal`	`integer`	The Tate-Chu-Yoko horizontal adjustment in points.
`TCY vertical`	`integer`	The Tate-Chu-Yoko vertical adjustment in points.
`text font`	<u>text</u>	The text font.
`titling`	`boolean`	If `true`, the OpenType titling alternates should be used.
`tracking`	`integer`	The tracking or range kerning amount in thousandths of an em.
`Tsume`	`real`	The percentage of space reduction around a Japanese character.
`underline`	`boolean`	If `true`, characters use underline style.
`vertical scale`	`real`	Character vertical scaling factor.

Property	Value type	What it is
warichu characters after break	integer	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu characters before break	integer	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu enabled	boolean	If true, Wari-Chu is enabled.
warichu gap	integer	The Wari-Chu line gap.
warichu justification	Valid values: auto justify center full justify last line center full justify full justify last line left full justify last line right left right	The Wari-Chu justification.
warichu lines	integer	The number of Wari-Chu (multiple text lines fit into a space meant for one) lines.
warichu scale	real	The Wari-Chu scale.

paragraph object commands

apply character style
apply paragraph style
change case
count
delete
deselect
duplicate
exists
make
move
select

Change hyphenation in text

The hyphenation of all text can be quickly changed from a script, as this example shows.

```
-- Enable hyphenation for every paragraph of the current document
tell application "Adobe Illustrator"
    if (count text frames of document 1) > 0 then
        set itemCounter to count text frames of document 1
        repeat with i from 1 to itemCounter
            set hyphenation of (every paragraph of text frame i of document 1) to true
        end repeat
    end if
end tell
```

Resize and justify paragraphs

```
-- Make a new document and a rectangle
-- Make an area-text text frame, assign the rectangle as it's path
-- Set contents of the text frame to text containing three paragraphs
-- Resize and justify the paragraphs
tell application "Adobe Illustrator"
    activate
    make new document
    make new rectangle in document 1 with properties {position:{100, 400}, width:400,
height:200}
    set areaText to make new text frame in document 1 with properties {kind:area text,
text path:the result}
    set theParagraph to "Left justified paragraph." & return & "Center justified
paragraph." & return & "Right justified paragraph."
    set the contents of areaText to theParagraph
    set the size of the text of areaText to 28
    set the justification of paragraph 1 of areaText to left
    set the justification of paragraph 2 of areaText to center
    set the justification of paragraph 3 of areaText to right
end tell
```

paragraph style, paragraph styles

A named style that remembers paragraph attributes.

NOTE: Paragraph attributes do not have default values, and they are undefined until explicitly set.

paragraph style object properties

Property	Value type	What it is
`best type`	type class	Read-only. The best type for the object's value.
`class`	type class	Read-only. The object's class.
`default type`	type class	Read-only. The default type for the object's value.
`aki left`	real	The amount of extra space (aki) added to the left side of each glyph in Japanese text (in thousandths of an em).
`aki right`	real	The amount of extra space (aki) added to the right side of each glyph in Japanese text (in thousandths of an em).
`alignment`	Valid values: bottom center icf bottom icf top roman baseline top	The character alignment type.
`alternate glyphs`	Valid values: default expert full width half width jis78 jis83 proportional width quarter width third width traditional jis90 jis04	Specifies the type of alternate glyphs.
`auto leading`	boolean	If `true`, automatic leading is used.
`auto leading amount`	real	The auto leading amount, as a percentage.

Property	Value type	What it is

baseline direction	Valid values:
	`standard`
	`Tate Chu Yoko`
	`vertical rotated`
		Specifies the Japanese text baseline direction.

baseline position	Valid values:
	`normal`
	`subscript`
	`superscript`
		The baseline position of text.

baseline shift	`real`	The amount of shift (in points) of the text baseline.
best type	type class	Read-only. The best type for the object's value.
BunriKinshi	`boolean`	If `true`, BunriKinshi is enabled.

Burasagari type	Valid values:
	`forced`
	`none`
	`standard`
		The Burasagari type which specifies whether punctuation is allowed to fall outside of the paragraph bounding box (not available when Kinsoku Shori is set to `none`).

capitalization	Valid values:
	`all caps`
	`all small caps`
	`normal`
	`small caps`
		The case of the text.

connection forms	`boolean`	If `true`, use the OpenType connection forms.
container	`reference`	Read-only. The object's container.
contextual ligature	`boolean`	If `true`, use the contextual ligature.
desired glyph scaling	`real`	Desired glyph scaling expressed as a percentage.
desired letter spacing	`real`	Desired letter spacing expressed as a percentage.
desired word spacing	`real`	Desired word spacing expressed as a percentage.
discretionary ligature	`boolean`	If `true`, use the discretionary ligature.
every line composer	`boolean`	If `true`, the *every line composer* is enabled.

Property	Value type	What it is

figure style	Valid values:
	  default
	  proportional
	  proportional oldstyle
	  tabular
	  tabular oldstyle
		Specifies which figure style to use in an OpenType font.

fill color	[color info](#)	The color of the text fill.
first line indent	real	First line left indent expressed in points.
fractions	boolean	If true, use the OpenType fractions.
horizontal scale	real	The character horizontal scaling factor expressed as a percentage (100 = 100%).
hyphenate capitalized words	boolean	If true, hyphenation is enabled for the capitalized words.
hyphenation	boolean	If true, hyphenation is enabled for the paragraph.
hyphenation preference	real	Hyphenation preference scale for better spacing (0) or fewer hyphens (1). Range: 0.0 to 1.0
hyphenation zone	real	Size of the hyphenation zone.
index	integer	Read-only. The index of this instance of the object.
italics	boolean	If true, the Japanese OpenType font supports italic text.

justification	Valid values:
	  center
	  full justify
	  full justify last line center
	  full justify last line left
	  full justify last line right
	  left
	  right
		Paragraph justification.

kerning method	Valid values:
	  auto
	  none
	  optical
		The automatic kerning method to use.

Kinsoku	Unicode text	The name of a Kinsoku Shori set (a set of characters which cannot be used to begin or end a line of Japanese text).

Property	Value type	What it is

`Kinsoku order`	Valid values:
	  push in
	  push out first
	  push out only
		The preferred Kinsoku order.

`KurikaeshiMojiShori`	boolean	If `true`, the Kurikaeshi Moji Shori is enabled (controls how repeated characters are handled in Japanese text).

`language`	Valid values:
	  Bokmal Norwegian
	  Brazillian Portuguese
	  Bulgarian
	  Canadian French
	  Catalan
	  Chinese
	  Czech
	  Danish
	  Dutch
	  Dutch 2005 Reform
	  English
	  Finnish
	  German 2006 Reform
	  Greek
	  Hungarian
	  Icelandic
	  Italian
	  Japanese
	  Nynorsk Norwegian
	  old German
	  Polish
	  Romanian
	  Russian
	  Spanish
	  Serbian
	  standard French
	  standard German
	  standard Portuguese
	  Swedish
	  Swiss German
	  Swiss German 2006 Reform
	  Turkish
	  UK English
	  Ukranian
		The language.

`leading`	real	Specifies the amount of space between two lines of text, in points.

`leading type`	Valid values:
	  Japanese
	  Roman
		Auto leading type.

`left indent`	real	Left indent of margin expressed in points.

Property	Value type	What it is
`ligature`	`boolean`	If `true`, use the ligature.
`maximum consecutive hyphens`	`integer`	Maximum number of consecutive hyphenated lines.
`maximum glyph scaling`	`real`	Maximum glyph scaling expressed as a percentage.
`maximum letter spacing`	`real`	Maximum letter spacing expressed as a percentage.
`maximum word spacing`	`real`	Maximum word spacing expressed as a percentage.
`minimum after hyphen`	`integer`	Minimum number of characters after a hyphen.
`minimum before hyphen`	`integer`	Minimum number of characters before a hyphen.
`minimum glyph scaling`	`real`	Minimum glyph scaling expressed as a percentage.
`minimum hyphenated word size`	`integer`	Minimum hyphenated word size.
`minimum letter spacing`	`real`	Minimum letter spacing expressed as a percentage.
`minimum word spacing`	`real`	Minimum word spacing expressed as a percentage.
`Mojikumi`	Unicode text	The name of a predefined Mojikumi set for Japanese text composition.
`name`	Unicode text	The paragraph style's name.
`no break`	`boolean`	If `true`, no line break is allowed.

`OpenType position`	Valid values:
	`default`
	`denominator`
	`numerator`
	`subscript`
	`superscript`
		The OpenType baseline position.

`ordinals`	`boolean`	If `true`, use the OpenType ordinals.
`ornaments`	`boolean`	If `true`, use the OpenType ornaments.
`overprint fill`	`boolean`	If `true`, overprint the fill of the text.
`overprint stroke`	`boolean`	If `true`, the stroke of the text may be overprinted.

Property	Value type	What it is
proportional metrics	boolean	If `true`, the proportional metrics in a Japanese OpenType font may be used.
right indent	real	Right indent of margin expressed in points.
roman hanging	boolean	If `true`, Roman hanging punctuation is enabled.
rotation	real	The character rotation angle.

single word justification	Valid values:
	`center`
	`full justify last line center`
	`full justify`
	`full justify last line left`
	`full justify last line right`
	`left`
	`right`
		Justification type for a single word.

size	real	Font size in points.
space after	real	Spacing after paragraph in points.
space before	real	Spacing before paragraph in points.
strike through	boolean	If `true`, characters use strike-through style.
stroke color	[color info]	The color of the text stroke.
stroke weight	real	line width of stroke.
stylistic alternates	boolean	If `true`, use the OpenType stylistic alternates.
swash	boolean	If `true`, use the OpenType swash.
tab stops	list of `tab stop info`	Tab stop settings.
TCY horizontal	integer	The Tate-Chu-Yoko horizontal adjustment in points.
TCY vertical	integer	The Tate-Chu-Yoko vertical adjustment in points.
text font	[text]	The text font.
titling	boolean	If `true`, use the OpenType titling alternates.
tracking	integer	The tracking or range kerning amount in thousandths of an em.
Tsume	real	The percentage of space reduction around a Japanese character.

Property	Value type	What it is
underline	`boolean`	If `true`, characters use underline style.
vertical scale	`real`	Character vertical scaling factor.
warichu characters after break	`integer`	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu characters before break	`integer`	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu enabled	`boolean`	If `true`, Wari-Chu is enabled.
warichu gap	`integer`	The Wari-Chu line gap.

warichu justification	Valid values:
	`auto justify`
	`center`
	`full justify last line center`
	`full justify`
	`full justify last line left`
	`full justify last line right`
	`left`
	`right`
		The Wari-Chu justification.

warichu lines	`integer`	The number of Wari-Chu (multiple text lines fit into a space meant for one) lines.
warichu scale	`real`	The Wari-Chu scale.

Apply paragraph styles

```
-- Make a new document and a rectangle
-- Make an area-text text frame, assign the rectangle as it's path
-- Set contents of text frame to text containing three paragraphs
-- Resize and justify the paragraphs
-- Make a new paragraph style with a set of options
-- Apply the paragraph style to the text of the text frame
tell application "Adobe Illustrator"
    activate
    make new document
    make new rectangle in document 1 with properties {position:{100, 400}, width:400,
height:200}
    set areaText to make new text frame in document 1 with properties {kind:area text,
text path:the result}
    set theParagraph to "Left justified paragraph." & return & "Center justified
paragraph." & return & "Right justified paragraph."
    set the contents of areaText to theParagraph
    set the size of the text of areaText to 28
    set the justification of paragraph 1 of areaText to left
    set the justification of paragraph 2 of areaText to center
    set the justification of paragraph 3 of areaText to right
    delay 2
    make new paragraph style in document 1 with properties {class:paragraph style,
name:"ParSty 1"}
    apply paragraph style paragraph style "ParSty 1" of document 1 to text of text frame
1 of document 1 with clearing overrides
end tell
```

path item, path items

A path or list of paths. A path is an art item such as those created using the Line, Rectangle, or Pen Tools. A path consists of path points that define its geometry. Path points are defined either as a `path point` object or as an x-y page coordinate pair.

The `path items` class gives you complete access to paths in Illustrator.

path item object elements

Element	Refer to by
path point	index, before/after, range, test

path item object properties

This object class inherits all properties from the `page item` class.

Property	Value type	What it is
area	real	Read-only. The area of this path in square points. An area may be negative or even 0. The paths winding order is determined by the sign of area. If the area is negative, the path is wound counter-clockwise. Self-intersecting paths may contain sub-areas that cancel each other out. Therefore, it is possible for a path's area to appear as zero even though it has apparent area.
clipping	boolean	If `true`, use this path as a clipping path.
closed	boolean	If `true`, this path closed.
entire path	list of path point info	All the path item's path points.
evenodd	boolean	If `true`, use the even-odd rule to determine insideness.
fill color	color info	The fill color of the path.
fill overprint	boolean	If `true`, the art beneath a filled object should be overprinted.
filled	boolean	If `true`, the path should be filled.
guides	boolean	If `true`, this path is a guide object.
length	real	Read-only. The length of this path in points.
polarity	Valid values: positive negative	The polarity of the path, used in the creation of compound paths.
resolution	real	The resolution of the path in dots per inch.
selected path points	list of object references	Read-only. All selected path points in the path.

Property	Value type	What it is

stroke cap	Valid values:
	`butted`
	`rounded`
	`projecting`
		The type of line capping.

stroke color	[color info](#)	The stroke color for the path.
stroke dash offset	`real`	The default distance into the dash pattern at which the pattern should be started
stroke dashes	list of `real` numbers	The lengths for dashes and gaps in dashed lines, starting with the first dash length, followed by the first gap length, and so on. Set to an empty list, `{}`, for a solid line.

stroke join	Valid values:
	`mitered`
	`rounded`
	`beveled`
		Type of join for the path.

stroke miter limit	`real`	When default stroke join is set to `mitered`, specifies when the join will be converted to beveled (squared-off) by default. The default miter limit of `4` means that when the length of the point reaches four times the stroke weight, the join switches from a miter join to a bevel join. Values: 1 to 500. 1 specifies a bevel join.
stroke overprint	`boolean`	If `true`, the art beneath the stroked object should be overprinted.
stroke width	`real`	The width of the stroke (in points).
stroked	`boolean`	If `true`, the path should be stroked.

path item object commands

[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[move](#)
[rotate](#)
[scale](#)
[transform](#)
[translate](#)

Setting stroke width and color

```
-- Set the stroke of the first path to a red 4 point line
tell application "Adobe Illustrator"
    if (count path items of document 1) > 0 then
        set properties of path item 1 of document 1 to ¬
            {stroke color:{red:255.0}, stroke width:4.0}
    end if
end tell
```

path point, path points

A point or points on a specific path. Each path point is made up of a fixed point (`anchor`) and a pair of handles (`left direction` and `right direction`). Any point can considered a corner point. Setting the `point type` property of a path point to a corner forces the left and right direction points to be on a straight line when the user attempts to modify them in the user interface.

path point object properties

Property	Value type	What it is
anchor	fixed point	The position of this point's anchor point.
best type	type class	Read-only. The best type for the path point object's value. Always returns `reference`.
class	type class	Read-only. The path point object's class, which is path point.
container	reference	Read-only. The object's container.
default type	type class	Read-only. The default type for the path point object's value. Always returns `reference`.
index	integer	Read-only. The position of this path point in the path item.
left direction	fixed point	The position of the path point's left direction point (in position).

point type	Valid values:
	   smooth
	   corner
		Is this a corner path point or a curve path point.

properties	record	All properties of this object returned as a record.
right direction	fixed point	The position of the path point's left direction point (out position).

selected	Valid values:
	   none
	   anchor selected
	   left selected
	   right selected
	   left right selected
		Specifies which points in this path point are currently selected.

path point object commands

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

Move a path point

```
-- Move the first point in a path to the same spot as the last point
tell application "Adobe Illustrator"
    set lastAnchor to ""
    if (count path items of document 1) > 0 then
        set lastAnchor to anchor of last path point of path item 1 of document 1
        set anchor of path point 1 of path item 1 of document 1 to lastAnchor
    end if
end tell
```

Get coordinates for path points

```
-- Returns the coordinates of each point on a path
tell application "Adobe Illustrator"
    if (count path items of document 1) > 0 then
        set anchorList to (anchor of every path point of path item 1 of document 1)
    end if
end tell
```

path point info

Path point information for a specific path item, returned by the `entire path` property of a **path item**. All path points in a specific path item can be retrieved and specified using `entire path`, which returns a list of path point info records.

path point info object properties

Property	Value type	What it is
anchor	list	The position of a `path point`'s anchor point.
left direction	list	The position of a `path point`'s left direction point (in position).
point type	Valid values: smooth corner	Specifies whether the point is a corner `path point` or a curve `path point`.
right direction	fixed point	The position of a `path point`'s left direction point (out position).

Get path point information

```
-- Returns the path points of the first path
tell application "Adobe Illustrator"
    if (count path items of document 1) > 0 then
        set pointList to entire path of path item 1 of document 1
    end if
end tell
```

pattern, patterns

An Illustrator pattern definition contained in a document. Patterns are shown in the Swatches palette. Each pattern is referenced by a [pattern color info](#) object, which defines the pattern's appearance.

pattern object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `pattern` object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `pattern`.
container	object reference	Read-only. A reference to the document that contains this `pattern`.
default type	type class	Read-only. The default type for the `pattern` object's value. Always returns `reference`.
index	integer	Read-only. The position of this `pattern` in the application.
name	Unicode text	The `pattern` name.
properties	record	All properties of this object returned as a record.

pattern object commands

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

Get the name of a pattern

```
-- Returns the name of the first pattern
tell application "Adobe Illustrator"
    set pathname to name of pattern 1 of document 1
end tell
```

pattern color info

A pattern color specification, used to specify a pattern color in conjunction with the `color` property. Pattern colors are created using a reference to an existing pattern in a document. A matrix may be specified to further transform the pattern color.

pattern color info object properties

This class inherits all properties from the color info class.

Property	Value type	What it is
matrix	matrix	An additional transformation matrix to manipulate the prototype `pattern`.
pattern	object reference	A reference to the `pattern` object that defines the pattern to use in this color definition.
reflect	boolean	If `true`, the prototype should be reflected before filling. Default: `false`
reflect angle	real	The axis (in degrees) around which to reflect. Default: 0.0
rotation	real	The angle (in degrees) to rotate the prototype pattern before filling. Default: 0.0
scale factor	fixed point	The horizontal and vertical scaling to scale the prototype pattern expressed as a fixed point. Default: 0.0
shear angle	real	The angle (in degrees) to slant the shear by. Default: 0.0
shear axis	real	The axis (in degrees) to be used for shearing. Default: 0.0
shift angle	real	The angle (in degrees) to translate the unscaled prototype `pattern` before filling. Default: 0.0
shift distance	real	The distance to translate the unscaled prototype `pattern` before filling. Default: 0.0

Using a pattern color

```
--Set the default fill of the document to the first pattern
tell application "Adobe Illustrator"
    set default fill color of document 1 to {pattern:pattern 1 of document 1}
end tell
```

CHAPTER 1: AppleScript Objects

PDF options   **137**

PDF options

Options that can be supplied when opening a PDF file.

PDF options object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
container	reference	Read-only. The object's container.
default type	type class	Read-only. The default type for the object's value.
page	integer	What page should be used when opening a multipage document. Default: 1
PDF crop bounds	Valid values: PDF art box PDF bleed box PDF bounding box PDF crop box PDF media box PDF trim box	What box should be used when placing a multipage document. Default: PDF media box
properties	record	All properties of this object returned as a record.

Open a PDF document

```
-- This function opens the file passed as
-- a file reference parameter at page 2, fileToOpen is
-- a reference to a multi-page PDF file and needs to
-- be set up before calling this function
on openMultipageFile(fileToOpen)
    tell application "Adobe Illustrator"
        set user interaction level to never interact
        set page of PDF file options of settings to 2
        open POSIX file fileToOpen as alias without dialogs
    end tell
end openMultipageFile
```

PDF save options

Options that can be supplied when saving a document as an Adobe PDF file. See the save command for additional details. This class contains properties used to specify options when saving a document to a PDF file. PDF save options can be supplied only in conjunction with the save command. It is not possible to get or create a PDF save options object.

Preset options can be exported from and imported to a document; see the export PDF preset and import PDF preset commands.

PDF save options object properties

Property	Value type	What it is
acrobat layers	boolean	Optional. Create Adobe Acrobat® layers from top-level layers; Acrobat 6 only option Default: `false`
allow printing	Valid values: `pdf 128 print high res` `pdf 128 print low res` `pdf 128 print none` `pdf 40 print high res` `pdf 40 print none`	Optional. PDF security printing permission. Default: `pdf 128 print high res`
artboard range	string	Optional. This is considered for multi-asset extraction, which specifies the artboard range. An empty string extracts all the artboards. Default: empty string
bleed link	boolean	Optional. Link four bleed values. Default: `true`
bleed offset	list	The bleed offset rectangle
changes allowed	Valid values: `pdf 128 any changes` `pdf 128 commenting allowed` `pdf 128 edit page allowed` `pdf 128 fill form allowed` `pdf 128 no changes` `pdf 40 any changes` `pdf 40 commenting allowed` `pdf 40 no changes` `pdf 40 page layout allowed`	Optional. Which PDF security changes are allowed. Default: `pdf 128 any changes`
color bars	boolean	Optional. Draw color bars. Default: `false`

Property	Value type	What it is

`color compression`	Valid values:
	`automatic JPEG high`
	`automatic JPEG low`
	`automatic JPEG maximum`
	`automatic JPEG medium`
	`automatic JPEG minimum`
	`automatic JPEG2000 high`
	`automatic JPEG2000 lossless`
	`automatic JPEG2000 low`
	`automatic JPEG2000 maximum`
	`automatic JPEG2000 medium`
		Optional. How color bitmap images should be compressed.
		Default: `automatic JPEG maximum`

`color conversion id`	Valid values:
	`repurpose`
	`color conversion to dest`
	`none`
		Optional. PDF color conversion policy.
		Default: `none`

`color destination id`	Valid values:
	`color dest doc cmyk`
	`color dest doc rgb`
	`color dest profile`
	`color dest working cmyk`
	`color dest working rgb`
	`none`
		Optional. The color destination, when color conversion is performed. Default: `none`

`color downsampling`	`real`	Optional. The resolution to which to downsample color image. If 0, no downsampling. Default: 150.
`color downsampling threshold`	`real`	Optional. Downsample if the image's resolution is above this value. Default: 450.0

`color profile id`	Valid values:
	`include all profiles`
	`include all rgb`
	`include dest profile`
	`leave profile unchanged`
	`none`
		Optional. PDF color profile inclusion policy.
		Default: `none`

`color resample`	Valid values:
	`average downsampling`
	`bicubic downsample`
	`nodownsample`
	`subsampling`
		Optional. How color bitmap images should be resampled. Default: `nodownsample`

`color tile size`	`integer`	Optional. Tile size when compressing with JPEG2000. Default: 256

`compatibility`	Valid values:
	`Acrobat 4`
	`Acrobat 5`
	`Acrobat 6`
	`Acrobat 7`
	`Acrobat 8`
		Optional. The version of the Acrobat file format to create. Default: `Acrobat 5`

`compress art`	`boolean`	Optional. If `true`, the line art and text should be compressed. Default: `true`

Property	Value type	What it is
`document password`	Unicode text	Optional. A password string to open the document. Default: no string
`enable access`	`boolean`	Optional. If `true`, accessing 128-bit should be enabled. Default: `true`
`enable copy`	`boolean`	Optional. If `true`, enable copying of text 128-bit. Default: `true`
`enable copy and access`	`boolean`	Optional. If `true`, enable copying and accessing 40-bit. Default: `true`
`enable plaintext`	`boolean`	Optional. If `true`, enable plaintext metadata 128-bit; available only for Acrobat 6. Default: `false`
`flattener preset`	Unicode text	Optional. The transparency flattener preset name.
`flattener settings`	[flattening options](#)	Optional. The printing flattener options.
`font subset threshold`	`real`	Optional. Include a subset of fonts when less than this percentage of characters are used. Range: 0.0 to 100.0. Default: 100.0
`generate thumbnails`	`boolean`	Optional. If `true`, generate thumbnails for the saved document. Default: `true`

`grayscale compression`	Valid values:
	`automatic JPEG high`
	`automatic JPEG low`
	`automatic JPEG maximum`
	`automatic JPEG medium`
	`automatic JPEG minimum`
	`automatic JPEG2000 high`
	`automatic JPEG2000 lossless`
	`automatic JPEG2000 low`
	`automatic JPEG2000 maximum`
	`automatic JPEG2000 medium`
	`automatic JPEG2000 minimum`
	`none`
		Optional. How grayscale bitmap images should be compressed. Default: `none`

`grayscale downsampling`	`real`	Optional. The resolution to which to downsample grayscale images. If 0, no downsampling. Default: 150.0
`grayscale downsampling threshold`	`real`	Optional. Downsample if the image's resolution is above this value. Default: 225.0

`grayscale resample`	Valid values:
	`average downsampling`
	`bicubic downsample`
	`nodownsample`
	`subsampling`
		Optional. How the grayscale bitmap images should be resampled. Default: `nodownsample`

Property	Value type	What it is
grayscale tile size	`integer`	Optional. Tile size when compressing with JPEG2000. Default: 256

monochrome compression	Valid values:
	  `CCIT3`
	  `CCIT4`
	  `none`
	  `run length`
	  `ZIP`
		Optional. How monochrome bitmap images should be compressed. Default: `none`

monochrome downsampling	`real`	Optional. The resolution to which to downsample monochrome images. If 0, no downsampling. Default: 300.0
monochrome downsampling threshold	`real`	Optional. Downsample if the image's resolution is above this value. Default: 450.0

monochrome resample	Valid values:
	  `average downsampling`
	  `bicubic downsample`
	  `nodownsample`
	  `subsampling`
		Optional. How monochrome bitmap images should be resampled. Default: `nodownsample`

offset	`real`	Optional. Custom offset (in points) for using the custom paper. Default: 0.0
optimization	`boolean`	Optional. If `true`, the PDF file should be saved for fast web view. Default: `false`
output condition	Unicode text	Optional. A comment that describes the intended printing condition. Default: no string
output condition id	Unicode text	Optional. The name of a registered printing condition. Default: no string
output intent profile	Unicode text	Optional. The color profile for the intended output. When CMS is on, this is the same as the profile selected for Destination in the Color group box. Default: no string
page info	`boolean`	Optional. If `true`, draw page information. Default: `false`

page marks style	Valid values:
	  `Japanese style`
	  `Roman`
		Optional. The page marks style.
		Default: `Roman`

PDF preset	Unicode text	Optional. Name of PDF preset to use. Maximum string length is 255 bytes.

Property	Value type	What it is

pdfXstandard	Valid values:
	  `PDFX None`
	  `PDFX 1a 2001`
	  `PDFX 1a 2003`
	  `PDFX 3 2002`
	  `PDFX 3 2003`
	  `PDFX 4 2007`
		Optional. The PDF standard, or none if not complying with any standard.
		Default: `PDFX None`

pdfXstandard description	Unicode text	Optional. A description of the selected PDF standard.
permission password	Unicode text	Optional. A password string to restrict editing security settings. Default: no string
preserve editability	`boolean`	Optional. If `true`, preserve Illustrator editing capabilities when saving the document. Default: `true`
printer resolution	`real`	Optional. Flattening style printer resolution. Default: 800.0
registration marks	`boolean`	Optional. If `true`, draw registration marks. Default: `false`
require doc password	`boolean`	Optional. If `true`, require a password to open the document. Default: `false`
require perm password	`boolean`	Optional. If `true`, a password is required to edit security settings. Default: `false`
trapped	`boolean`	Optional. If `true`, manual trapping has been prepared in the document. Default: `false`

trim mark weight	Valid values:
	  `trimmarkweight0125`
	  `trimmarkweight025`
	  `trimmarkweight05`
		Optional. Weight of the trim marks. Default:
		`trimmarkweight0125`

trim marks	`boolean`	Optional. If `true`, draw trim marks. Default: `false`
view pdf	`boolean`	Optional. If `true`, view PDF after saving. Default: `false`

Save to PDF

This handler processes a folder of Illustrator files, saving each file as a PDF file, with Illustrator editability and Acrobat® 6 compatibility. Note that the `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destFolder is an alias to a folder where the files are to be saved

on SaveFilesAsPDF(fileList, filePath, destFolder)
    set destPath to destFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destPath & fileName & ".pdf"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                save current document in file newFilePath as pdf ¬
                    with options {class:PDF save options ¬
                    , compatibility:Acrobat 5 ¬
                    , preserve editability:true}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsPDF
```

Photoshop export options

Options that can be supplied when exporting a document as a Photoshop file. See the export command for additional details.

This class contains properties that specify options when exporting a document as a Photoshop file. `Photoshop export options` can only be supplied in conjunction with the `export` command. It is not possible to get or create a `Photoshop export options` object.

Photoshop export options object properties

Property	Value type	What it is
antialiasing	boolean	If `true`, the exported image should be anti-aliased. Default: `true`
artboard range	string	If `save multiple artboards` is `true`, this is considered for multi-asset extraction, which specifies the artboard range. An empty string extracts all the artboards. Default: empty string
color space	Valid values: Gray RGB CMYK	The color space of the exported file. Default: `RGB`
editable text	boolean	If `true`, text objects should be exported as editable text layers. Default: `true`
embed ICC profile	boolean	If `true`, an ICC profile should be embedded in the exported image. Default: `false`
maximum editability	boolean	If `true`, preserve as much of the original document's structure as possible. Default: `true`
resolution	real	Specifies the resolution of the exported image in dots per inch. Default: 150.0
save multiple artboards	boolean	If `true`, all artboards or range of artboards are saved. Default: `false`
warnings	boolean	If `true`, a warning dialog should be displayed because of conflicts in the export settings. Default: `true`
write layers	boolean	If `true`, the layers of the Illustrator document should be preserved in the exported image. Default: `true`

Export to Photoshop format with options

This handler saves all files in a folder as layered Photoshop files. Note that the `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destFolder is an alias to a folder where the files are to be saved

on SaveFilesAsPhotoshop(fileList, filePath, destFolder)
    set destPath to destFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destPath & fileName & ".psd"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                export current document to file newFilePath as Photoshop ¬
                    with options {class:Photoshop export options ¬
                    , color space:RGB ¬
                    , embed ICC profile:true ¬
                    , resolution:150}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsPhotoshop
```

Photoshop options

You can supply options when opening a Photoshop file. See the open command in the command reference for additional details.

Photoshop options object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
container	specifier	Read-only. The object's container.
default type	type class	Read-only. The default type for the object's value.
layer comp	Unicode text	The name of the layer comp to use when the document is converted.
preserve hidden layers	boolean	If true, preserve hidden layers when the document is converted. Default: false.
preserve image maps	boolean	If true, image maps should be preserved when the document is converted. Default: true
preserve layers	boolean	If true, layers should be preserved when the document is converted. Default: true
preserve slices	boolean	If true, slices should be preserved when the document is converted. Default: true
properties	record	All properties of this object returned as a record.

Open a Photoshop file

```
-- This function opens the passed in Photoshop file with
-- open options to preserve layers and correct aspect ratio
-- set to false, fileToOpen is the file reference and needs
-- to be set up before calling this function
on PhotoshopFileOptions(fileToOpen)
    tell application "Adobe Illustrator"
        set user interaction level to never interact
        activate
        set photoshopOptions to {class:Photoshop options, preserve layers:true,
artboardRowsOrCols:false}
        set IllustratorPreferences to {class:Illustrator preferences, Photoshop file
options:photoshopOptions}
        open POSIX file fileToOpen as alias without dialogs
    end tell
end PhotoshopFileOptions
```

placed item, placed items

An artwork item placed in a document as a linked file. Users can place files with the **File > Place** command in Illustrator. Placed items can be created in a script using the technique illustrated in the following example.

placed item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
bounding box	rectangle	Read-only. Dimensions of `placed item` regardless of transformations.
content variable	anything	The content variable to which this `placed item` is bound. It is not necessary to set the type of the `content variable` before binding. Illustrator automatically sets the type to `image`.
file path	file specification	The file containing the placed artwork.
matrix	matrix	The transformation matrix applied to the `placed item`.
properties	record	All properties of this object returned as a record.

placed item object commands

count
delete
duplicate
embed
exists
make
move
relink
rotate
scale
trace placed
transform
translate

Place a file in a document

```
-- This function adds a new placed item to a document from a file reference,
-- fileToPlace, which is passed in during the function call, fileToPlace is an
-- alias or file reference to an art file, which must be set up before calling this
-- function, itemPosition is a fixed point at which to position the placed item
on PlacedItemCreate(fileToPlace)
    tell application "Adobe Illustrator"
        set itemPosition to {100.0, 200.0}
        set placedRef to make new placed item in document 1 ¬
            with properties {file path:fileToPlace, position:itemPosition}
    end tell
end PlacedItemCreate
```

plugin item, plugin items

An art item or objects created by an Illustrator plug-in. Scripts cannot create plug-in items, but can duplicate, copy, and paste them.

plugin item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
properties	record	All properties of this object returned as a record.
is tracing	boolean	Read-only. If `true`, this plugin group was created by tracing a raster art item.
tracing	tracingobject	Read-only. If this object was created by tracing a raster art item, the `tracingobject` that associates the resulting vector art with tracing options. Use the <u>expand tracing</u> and <u>release tracing</u> commands with this object to convert this plugin group to a `group item`, or to revert to the original raster art.

plugin item object commands

<u>count</u>
<u>delete</u>
<u>duplicate</u>
<u>exists</u>
<u>move</u>
<u>rotate</u>
<u>scale</u>
<u>transform</u>
<u>translate</u>

PNG8 export options

Options that can be supplied when exporting a document as a PNG file with 8-bit color. See the <u>export</u> command for additional details.

This class contains properties that specify options when exporting a document as a PNG8 file. `PNG8 export options` can only be supplied in conjunction with the `export` command. It is not possible to get or create a `PNG8 export options` object.

PNG8 export options object properties

Property	Value type	What it is
antialiasing	boolean	If `true`, the resulting image should be anti-aliased. Default: `true`
artboard clipping	boolean	If `true`, the resulting image should be clipped to the artboard. Default: `false`
color count	integer	The number of colors in the exported color table. This value can range from 2 to 256. The default value is 128 if the property is not set explicitly.

color dither	Valid values:
	    diffusion
	    pattern dither
	    noise
	    none
		The method used to dither colors. Default: `diffusion`

color reduction	Valid values:
	    selective
	    adaptive
	    perceptual
	    web
		The method used to reduce the number of colors in the document. Default: `selective`

dither percent	integer	How much should the colors be dithered as a percentage. Range: 0 to 100. Default: 88
horizontal scaling	real	The percentage horizontal scaling factor to apply to the resulting image. Range: 0.0 to 100.0 Default: 100.0
interlaced	boolean	If `true`, the resulting image be should be interlaced. Default: `false`
matte	boolean	If `true`, the artboard should be matted with a color. Default: `true`
matte color	<u>RGB color info</u>	The color to use when matting the artboard. Default: `white`
saving as HTML	boolean	If `true`, the resulting image should be saved with an accompanying HTML file. Default: `false`
transparency	boolean	If `true`, the resulting image should use transparency. Default: `true`

Property	Value type	What it is
vertical scaling	real	The percentage vertical scaling factor to apply to the resulting image. Range: 0.0 to 100.0. Default: 100.0
web snap	integer	How much should the color table be changed to match the web pallet as a percentage. Range: 0 to 100. Default: 0

Export to PNG8

This handler saves all files in a folder as 8-bit PNG files in HTML format with dithering and interlacing. The `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destFolder is an alias to a folder where the files are to be saved

on SaveFilesAsPNG8HTML(fileList, filePath, destFolder)
    set destPath to destFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destPath & fileName & ".png"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                export current document to file newFilePath as PNG8 ¬
                    with options {class:PNG8 export options ¬
                    , color count:64 ¬
                    , color reduction:web ¬
                    , color dither:pattern dither ¬
                    , dither percent:50 ¬
                    , interlaced:true}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsPNG8HTML
```

PNG24 export options

Options that can be supplied when exporting a document as a PNG file with 24-bit color. See the export command for additional details.

This class contains properties that specify options to be used when exporting a document as a PNG24 file. `PNG24 export options` can only be supplied in conjunction with the export command. It is not possible to get or create a `PNG24 export options` object.

PNG24 export options object properties

Property	Value type	What it is
`antialiasing`	`boolean`	If `true`, the resulting image should be anti-aliased. Default: `true`
`artboard clipping`	`boolean`	If `true`, the resulting image should be clipped to the artboard. Default: `false`
`horizontal scaling`	`real`	The percent horizontal scaling factor to apply to the resulting image. Range: 0.0 to 100.0. Default: 100.0
`matte`	`boolean`	If `true`, the artboard should be matted with a color. Default: `true`
`matte color`	RGB color info	The color to use when matting the artboard. Default: {255.0, 255.0, 255.0}
`saving as HTML`	`boolean`	If `true`, the resulting image be saved with an accompanying HTML file. Default: `false`
`transparency`	`boolean`	If `true`, the resulting image should use transparency. Default: `true`
`vertical scaling`	`real`	The percentage vertical scaling factor to apply to the resulting image. Range: 0.0 to 100.0. Default: 100.0

Exporting to PNG24

This handler saves all files in a folder as 24-bit PNG files in HTML format scaled to 50%. Note that the `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destFolder is an alias to a folder where the files are to be saved

on SaveFilesAsPNG24(fileList, filePath, destFolder)
    set destPath to destFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destPath & fileName & ".png"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                export current document to file newFilePath as PNG24 ¬
                    with options {class:PNG24 export options ¬
                    , horizontal scaling:50.0 ¬
                    , vertical scaling:50.0 ¬
                    , saving as HTML:false}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsPNG24
```

polygon

A class used to create a multi-sided path item in an Illustrator document. This object is available only in the context of a `make` command, which creates an instance of the `path item` class. This special class allows you to quickly create complex path items using the properties provided. Properties usually associated with path items, such as `fill color`, can also be specified at the time of creation.

If you do not specify any properties when making a new polygon, default values are used.

polygon object properties

Property	Value type	What it is
`center point`	`fixed point`	Write-once. The center point for the polygon. Default: {200.0, 300.0}
`radius`	`real`	Write-once. The radius of the polygon's points. Default: 50.0
`reversed`	`boolean`	Write-once. If `true`, the polygon path is reversed. Default: `false`
`sides`	`integer` (unsigned)	Write-once. The number of sides for the polygon. Default: 8

polygon object commands

make

Create a polygon

```
-- Make an octagon in document 1
tell application "Adobe Illustrator"
    set pathRef to make new polygon in document 1 with properties ¬
        {center point:{200.0, 200.0}, radius:40.0, sides:8}
end tell
```

postscript options

Specifies the options for printing to a PostScript language printer or image setter when printing a document with the <u>print</u> command.

postscript options object properties

Property	Value type	What it is
`binary printing`	`boolean`	If `true`, job is to be printed in binary mode. Default: `false`
`compatible shading`	`boolean`	If `true`, use PostScript language level 1 compatible gradient and gradient mesh printing. Default: `false`
`force continuous tone`	`boolean`	If `true`, force continuous tone. Default: `false`

`image compression`	Valid values:
	`JPEG`
	`none`
	`RLE`
		The image compression type. Default: `none`

`negative printing`	`boolean`	If `true`, print in negative mode. Default: `false`

`PostScript`	Valid values:
	`level 1`
	`level 2`
	`level 3`
		The PostScript language level. Default: `level 2`

`shading resolution`	`real`	The shading resolution in dots per inch. Range: 1.0 to 9600.0;.Default: 300.0

PPD file

Associates properties with a PPD file to be used in printing to a PostScript language printer or image setter. The properties are not available unless a document is open.

PPD file object properties

Property	Value type	What it is
name	Unicode text	The PPD model name.
properties	PPD properties	The PPD file information.

Save to PPD

```
-- Make a new document
-- Get the PPDs
-- Get the name, PS Level, and file path of the first PPD
-- Make a new text frame with the PPD info as its contents
tell application "Adobe Illustrator"
    activate
    make new document
    set PPDFiles to PPDs
    set PPDName to name of item 1 of PPDFiles
    set PPDProperties to get properties of item 1 of PPDFiles
    set PPDLevel to language level of PPDProperties
    set PPDPath to file path of PPDProperties
    set textContents to PPDName & return & "PostScript Level " & PPDLevel & return & "PPD
Path: " & PPDPath as string
    make new text frame in document 1 with properties {contents:textContents,
position:{20, 600}}
end tell
```

PPD properties

Specifies information about a PPD file.

PPD properties object properties

Property	Value type	What it is
`file path`	File object	Path specification for the PPD file.
`language level`	Unicode text	The PostScript language level.
`screens`	list of [separation screen](#)	List of color separation screens.
`spot functions`	list of [screen spot function](#)	List of color separation screen spot functions.

Using PPD information

```
-- Make a new document
-- Get the PPD files
-- Get name, PS Level, screens, screen spot functions, and file path of first PPD
-- For each screen, get the name, angle, and frequency
-- For each spot function, get the name and the function
-- Make a new text frame with the PPD info as its contents
tell application "Adobe Illustrator"
    activate
    make new document
    set PPDFiles to PPDs
    set PPDName to name of item 1 of PPDFiles
    set PPDProperties to get properties of item 1 of PPDFiles
    set PPDLevel to language level of PPDProperties
    set PPDPath to file path of PPDProperties
    set PPDScreens to screens of PPDProperties
    set screensText to "Screens" & return
    repeat with PPDScreen in PPDScreens
        set PPDScreenName to name of PPDScreen
        set PPDScreenAngle to angle of properties of PPDScreen
        set PPFScreenFrequency to frequency of properties of PPDScreen
        set screensText to screensText & tab & PPDScreenName & ¬
            " - Angle: " & PPDScreenAngle & ", Frequency: " & PPFScreenFrequency ¬
            & return as string
    end repeat
    set PPDSpotFunctions to spot functions of PPDProperties
    set PPDSpotFunctionText to "Spot Functions" & return
    repeat with PPDSpotFunction in PPDSpotFunctions
        set PPDSpotFunctionName to name of PPDSpotFunction
        set PPDSpotFunctionTX to spot function of PPDSpotFunction
        set PPDSpotFunctionText to PPDSpotFunctionText & tab ¬
            & PPDSpotFunctionName & ": " & PPDSpotFunctionTX ¬
            & return as string
    end repeat
    set textContents to PPDName & return & ¬
        "PostScript Level " & PPDLevel & return & "PPD Path: " & PPDPath & return &
return ¬
        & screensText & return & return & PPDSpotFunctionText as string
    make new text frame in document 1 ¬
```

```
            with properties {contents:textContents, position:{20, 700}}
    end tell
```

print options

Collects all print options when printing a document with the [print](#) command.

print options object properties

Property	Value type	What it is
color management settings	[color management options](#)	The printing color management options.
color separation settings	[color separation options](#)	The printing color separation options.
coordinate settings	[coordinate options](#)	The printing coordinate options.
flattener preset	Unicode text	The transparency flattener preset name.
flattener settings	[flattening options](#)	The printing flattener options.
font settings	[font options](#)	The printing font options.
job settings	[job options](#)	The printing job options.
page marks settings	[page marks options](#)	The printing page marks options.
paper settings	[paper options](#)	The paper options.
postscript settings	[postscript options](#)	The printing PostScript options.
PPD name	Unicode text	The name of the PPD file.
print preset	Unicode text	The name of the printer preset to use.
printer name	Unicode text	The printer name.

Print with options

```
-- Make new document. add symbol items
-- Set job options, color management options, coordinate options, flattening options
-- Print the document using these options
tell application "Adobe Illustrator"
    activate
    make new document
    repeat with i from 1 to (count of symbols in document 1)
        round (i / 2 - (round (i / 2) rounding down)) rounding up
        make new symbol item in document 1 with properties ¬
            {symbol:symbol i of document 1, position:{100 + (the result * 150), (50 + i *
70)}} ¬

    end repeat
    set jobOptions to {class:job options, designation:all layers, reverse pages:true} ¬

    set colorOptions to {class:color management options, name:"ColorMatch RGB",
intent:saturation} ¬

    set coordinateOptions to {class:coordinate options, fit to page:true}
    set flatteningOptions to ¬
        {class:flattening options, clip complex regions:true, gradient resolution:60,
rasterization resolution:60} ¬

    set printOptions to ¬
        ¬
            {class:print options, job settings:jobOptions, color management
settings:colorOptions, coordinate settings:coordinateOptions, flattener
settings:flatteningOptions} ¬

    print document 1 options printOptions
end tell
```

printer

Associates an installed printer with a printer configuration object.

printer object properties

Property	Value type	What it is
name	Unicode text	The printer name.
properties	[printer properties](#)	The printer information.

Listing printers

```
-- Make a new documet
-- Get the name of every printer
-- Display the list of names
tell application "Adobe Illustrator"
    set printerList to ""
    activate
    make new document
    set textRef to make new text frame in current document
    if printers is not {} then
        name of every item of printers as list
        repeat with theName in the result
            set printerList to printerList & theName & return
        end repeat
        set theText to printerList
        set position of textRef to {200, 600}
    else
        set theText to "No installed printers"
    end if
    set contents of textRef to theText
end tell
```

printer properties

Specifies configuration information for a printer.

printer properties object properties

Property	Value type	What it is
`binary printing`	`boolean`	If `true`, the printer supports binary printing.

`color support`	Valid values:
	   `black and white output`
	   `color output`
	   `grayscale output`
		The printer's color capability.

`custom paper sizes`	`boolean`	If `true`, the printer supports custom paper sizes.
`custom paper transverse`	`boolean`	If `true`, the printer supports custom paper transverse.
`default resolution`	`real`	The printer's default resolution. Minimum: 0.0
`InRIP separation support`	`boolean`	If `true`, the printer supports InRIP color separation.
`maximum height offset`	`real`	The custom paper's maximum height offset.
`maximum paper height`	`real`	Custom paper's maximum height.
`maximum paper width`	`real`	Custom paper's maximum width.
`maximum resolution`	`real`	The printer's maximum device resolution. Minimum: 0.0
`maximum width offset`	`real`	Custom paper's maximum width offset.
`minimum height offset`	`real`	Custom paper's minimum height offset.
`minimum paper height`	`real`	Custom paper's minimum height.
`minimum paper width`	`real`	Custom paper's minimum width.
`minimum width offset`	`real`	Custom paper's minimum width offset.
`paper sizes`	list of paper	A list of supported paper sizes.

Property	Value type	What it is

PostScript	Valid values:
	`level 1`
	`level 2`
	`level 3`
		The PostScript language level.

printer type	Valid values:
	`non PostScript printer`
	`PostScript printer`
	`unknown`
		The type of printer.

raster effect options

Specifies raster effects settings for the document. All properties are optional.

raster effect options object properties

Property	Value type	What it is
`antialiasing`	`boolean`	If `true`, the image is antialiased. Default: `false`
`clipping mask`	`boolean`	If `true`, a clipping mask is created for the image. Default: `false`

`color model`	Valid values:
	    `default rasterization`
	    `grayscale rasterization`
	    `bitmap rasterization`
		The color model for the rasterization. Default: `default rasterization`

`convert spot colors`	`boolean`	If `true`, all spot colors are converted to process colors for the image. Default: `false`
`padding`	`real`	The amount of white space (in points) added around the object during rasterization. Default: .0
`resolution`	`real`	The resolution in dots per inch (dpi). Range: 72.0 to 2400.0. Default: 300.0
`transparency`	`boolean`	If `true`, the image uses transparancy. Default: `false`

raster item, raster items

A bitmap art item or list of objects. You can create `raster item`s from a script if you use an external file. You can also create new raster items by duplicating or copying and pasting an existing `raster item`.

raster item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
bits per channel	`integer`	Read-only. The number of bits per channel.
bounding box	`rect`	The dimensions of the `raster item` regardless of transformations.
channels	`integer`	Read-only. The number of channels.

color space	Valid values:
	`Gray`
	`RGB`
	`CMYK`
		Read-only. The color space of the `raster item`.

colorants	list of Unicode text	Read-only. The colorant used in the raster art.
colorized	`boolean`	Read-only. If true, the raster art is a colorized grayscale image.
content variable	anything	The contents of the variable to which this raster item is bound. It is not necessary to set the type of the `content variable` before binding. Illustrator automatically sets the type to image.
embedded	`boolean`	If `true`, the `raster item` is embedded within the illustration.
file path	file specification	Read-only. The file containing the `raster item`, if it is stored externally.
matrix	`matrix`	The transformation matrix of the raster art item.
overprint	`boolean`	If true, the raster art is overprinting.
properties	`record`	All properties of this object returned as a record.

status	Valid values:
	`no data`
	`data from file`
	`modified data`
		Read-only. The status of the linked image, if the image is stored externally.

transparent	`boolean`	If true, the raster art is transparent.

raster item object commands

[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[move](#)
[rotate](#)
[scale](#)
[trace raster](#)
[transform](#)
[translate](#)

Create a raster item

```
-- This handler accepts filePath as the path to a new
-- raster item and embeds the item in to a new document
on rasterItemCreate(filePath)
    tell application "Adobe Illustrator"
        set myDoc to make new document
        set myPosition to {0.0, height of myDoc}
        set myPlacedItem to make new placed item in myDoc with properties ¬
            {file path:filePath, position:myPosition}
        embed myPlacedItem
    end tell
end rasterItemCreate
```

rasterize options

Specifies options that may be supplied when rasterizing artwork. All properties are optional.

rasterize options object properties

Property	Value type	What it is

`antialiasing method`	Valid values:
	    `none`
	    `art optimized`
	    `type optimized`
		The type of antialiasing method.
		Default: `art optimized`

`background black`	`boolesn`	If `true`, the rasterization is done against a black background (intead of white). Default: `false`
`clipping mask`	`boolean`	If `true`, a clipping mask is created for the image. Default: `false`

`color model`	Valid values:
	    `default rasterization`
	    `grayscale rasterization`
	    `bitmap rasterization`
		The color model for the rasterization.
		Default: `default rasterization`

`convert spot colors`	`boolean`	If `true`, spot colors are converted to process colors for the image. Default: `false`
`convert text to outlines`	`boolean`	If `true`, all text is converted to outlines before rasterization. Default: `false`
`include layers`	`boolean`	If `true`, the resulting image incorporates layer attributes (like opacity and blend mode). Default: `false`
`padding`	`real`	The amount of white space (in points) added around the object during rasterization. Default: .0
`resolution`	`real`	The rasterization resolution in dots per inch (dpi). Range: 72.0 to 2400.0. Default: 300.0
`transparency`	`boolean`	If `true`, the image uses transparency. Default: `false`

rectangle

A class used to create a rectangular path in an Illustrator document. This object is available only in the context of a `make` command, which creates an instance of the `path item` class. This special class allows you to quickly create complex path items. Properties associated with `path items`, such as `fill color` and `note`, can also be specified at the time of creation.

A rectangle is stored as a list of four real numbers, where the first item is the leftmost horizontal coordinate of the rectangle, the second item is the top vertical coordinate of the rectangle, the third item is the rightmost horizontal coordinate, and the fourth item is the bottom vertical coordinate of the rectangle.

In the Illustrator coordinate system, vertical coordinates increase from bottom to top, which is the opposite of screen coordinates. This mean that the top coordinate value in a rectangle is larger than the bottom coordinate value.

rectangle object properties

Property	Value type	What it is
bounds	list	Write-once. The bounds of the rectangle. Default: {100.0, 200.0, 175.0, 100.0}
reversed	boolean	Write-once. If `true`, the path is reversed. Default: `false`

rectangle object commands

[make](#)

Creating a rectangle

```
-- Make a square in document 1
tell application "Adobe Illustrator"
   make new document
   set pathRef to make new rectangle at beginning of document 1 ¬
      with properties {bounds:{50.0, 200.0, 200.0, 50.0}, note:"square"}
end tell
```

Using rectangle values

The values in a `rectangle` can be used in a number of ways in a script.

```
tell application "Adobe Illustrator"
    -- Get the bounds of a page item
    set itemBounds to geometric bounds of page item 1 of document 1
    -->  for example: {100.0, 400.0, 300.0, 200.0}
    -- Assigns the four values in a rectangle point to four variables
    set {leftBound, topBound, rightBound, bottomBound} to itemBounds
    -- or assign to four variables directly
    set {leftBound, topBound, rightBound, bottomBound} to geometric bounds of page item
1 of document 1

    -- Calculate center of page item from its bounds
    set xCenter to ((item 1 of itemBounds) + (item 3 of itemBounds)) / 2
    set yCenter to ((item 2 of itemBounds) + (item 4 of itemBounds)) / 2
    --> example: xCenter = 200.0, yCenter = 300.0

    -- or calculate the center using the individual coordinate variables
    set xCenter to (leftBound + rightBound) / 2
    set yCenter to (topBound + bottomBound) / 2

    -- Change the left value in a fixed rectangle
    set item 1 of itemBounds to (item 1 of itemBounds) + 100.0
    -->  example: {200.0, 400.0, 300.0, 200.0}
end tell
```

RGB color info

An RGB color specification, used to specify a RGB color where a `color info` object is required.

If the color space of a document is CMYK and you specify the color value for a page item in that document using `RGB color info`, Illustrator will translate the RGB color specification into a CMYK color specification. The same thing happens if the document's color space is RGB and you specify colors using `CMYK color info`. Since this translation can cause information loss you should specify colors using the color info class that matches the document's color space.

RGB color info object properties

This class inherits all properties from the <u>color info</u> class.

Property	Value type	What it is
red	real	The red color value. Range: 0.0 to 255.0. Default: 0.0
green	real	The green color value. Range: 0.0 to 255.0. Default: 0.0
blue	real	The blue color value. Range: 0.0 to 255.0. Default: 0.0

Set the default stroke color to an RGB color

```
-- Set the default stroke color of document 1 to yellow
tell application "Adobe Illustrator"
    set default stroke color of document 1 to {red:255, green:255, blue:0}
end tell
```

rounded rectangle

A class used to create a rectangular path with rounded corners in an Illustrator document. This object is available only in the context of a `make` command, which creates an instance of the `path item` class. This special class allows you to quickly create complex path items. Properties associated with `path items`, such as `fill color` and `note`, can also be specified at the time of creation.

If you do not specify any properties when making a new rounded rectangle, default values are used.

rounded rectangle object properties

Property	Value type	What it is
bounds	rect	Write-once. The bounds of the rectangle to create. Default: {100.0, 100.0, 150.0, 200.0}
horizontal radius	real	Write-once.The horizontal radius of the rectangle's rounded corners. Default: 15.0
reversed	boolean	Write-once.If `true`, the rectangle path is reversed. Default: `false`
vertical radius	real	Write-once. The vertical radius of the rectangle's rounded corners. Default: 20.0

rounded rectangle object commands

[make](#)

Create a rounded rectangle

```
-- Make a rounded rectangle
tell application "Adobe Illustrator"
    make new document
    set pathRef to make new rounded rectangle in document 1 with properties ¬
        {bounds:{50.0, 200.0, 200.0, 50.0}, horizontal radius:20.0, vertical
radius:25.0}
end tell
```

screen properties

Contains screen information.

screen properties object properties

Property	Value type	What it is
angle	real	The screen's angle in degrees.
default screen	boolean	If `true`, it is the default screen.
frequency	real	The screen's frequency.

Get screen properties

```
-- PPD Screens
-- Make a new document, get the PPDs
-- Get the name, and screens of the first PPD
-- For each screen, get the name, angle, and frequency
-- Display the results of the PPD info in a text frame

tell application "Adobe Illustrator"
    activate
    make new document
    set PPDFiles to PPDs
    set PPDName to name of item 1 of PPDFiles
    set PPDProperties to get properties of item 1 of PPDFiles
    set PPDScreens to screens of PPDProperties
    set screensText to "Screens" & return
    repeat with PPDScreen in PPDScreens
        set PPDScreenName to name of PPDScreen
        set PPDScreenAngle to angle of properties of PPDScreen
        set PPFScreenFrequency to frequency of properties of PPDScreen
        set screensText to screensText & tab & PPDScreenName & ¬
            " - Angle: " & PPDScreenAngle & ", Frequency: " & PPFScreenFrequency ¬
            & return as string
    end repeat
    set textContents to PPDName & return & screensText
    make new text frame in document 1 ¬
        with properties {contents:textContents, position:{20, 600}}
end tell
```

screen spot function

Information about the color separation screen spot function.

screen spot function object properties

Property	Value type	What it is
`name`	Unicode text	The color separation `screen spot function` name.
`spot function`	Unicode text	The spot function in terms of the PostScript commands.

Get screen spot function information

```
-- PPD Screen Spot Functions
-- Make a new document, get the PPDs
-- Get the name, and spot functions of the first PPD
-- For each spot function, get the name and the function
-- Display the results of the PPD info in a text frame

tell application "Adobe Illustrator"
    activate
    make new document
    set PPDFiles to PPDs
    set PPDName to name of item 1 of PPDFiles
    set PPDProperties to get properties of item 1 of PPDFiles
    set PPDSpotFunctions to spot functions of PPDProperties
    set PPDSpotFunctionText to "Spot Functions" & return
    repeat with PPDSpotFunction in PPDSpotFunctions
        set PPDSpotFunctionName to name of PPDSpotFunction
        set PPDSpotFunctionTX to spot function of PPDSpotFunction
        set PPDSpotFunctionText to PPDSpotFunctionText & tab ¬
            & PPDSpotFunctionName & ": " & PPDSpotFunctionTX ¬
            & return as string
    end repeat
    set textContents to PPDName & return & PPDSpotFunctionText
    make new text frame in document 1 ¬
        with properties {contents:textContents, position:{20, 600}}
end tell
```

separation screen

Represents a color-separation screen.

separation screen object properties

Property	Value type	What it is
`name`	Unicode text	The color-separation screen name.
`properties`	screen properties	The color-separation screen information.

spot, spots

A custom color definition, or list of definitions, contained in a document.

If no properties are specified when creating a new spot, default properties will be provided. If specifying the color, however, you must use the same color space as the document, either CMYK or RGB; otherwise, an error will result. When created, the spot is added to the end of the swatches list in the Swatches palette.

spot object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `spot` object. Always returns `reference`.
class	type class	Read-only. The object's class, which is `spot`.
color	[spot color info](#)	The color information for this spot color.

color type	Valid values:
	　`process color`
	　`registration color`
	　`spot color`
		The color model for the spot color.

container	object reference	Read-only. A reference to the document that contains this spot color.
default type	type class	Read-only. Default type for the `spot`. Always returns `reference`.
index	`integer`	Read-only. The position of this spot in the document.
name	Unicode text	The spot color's unique name.
properties	`record`	All properties of this object returned as a record.

spot kind	Valid values:
	　`spot cmyk color`
	　`spot rgb color`
	　`spot lab color`
		Read-only. The kind of spot color (RGB, CMYK, or LAB). This is the name of the color kind contained in the `spot` object.

spot object commands

[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[get internal color](#)
[make](#)

Create a spot color

```
-- Make a new spot with name and color properties
tell application "Adobe Illustrator"
    set spotColorCount to count of spots in document 1
    -- set up the appropriate color record for the document color space
    set docColorSpace to color space of document 1
    if (docColorSpace is CMYK) then
        set newSpotColor to ¬
            {cyan:25.0, magenta:75.0, yellow:0.0, black:0.0}
    else
        set newSpotColor to {red:255.0, green:0.0, blue:25.0}
    end if
    -- now create the new spot
    make new spot in document 1 with properties ¬
        {name:"My Spot", color:newSpotColor}
end tell
```

spot color info

A spot color specification, used to specify a spot color in the `spot` object's `color` property.

spot color info object properties

This class inherits all properties from the [color info](#) class.

Property	Value type	What it is
spot	object reference	A reference to the `spot` object which defines the color. Must be set to a reference to an existing spot color definition
tint	`real`	The tint of the color. Range: 0.0 to 100.0. Default: 100.0

Setting the default stroke color to a spot color

```
-- Make a new spot color and apply a 50% tint to the default stroke color
tell application "Adobe Illustrator"
    -- create a document with RGB color space
    make new document with properties {color space:RGB}
    set spotColorCount to count of spots in document 1
    set newSpot to make new spot in document 1 with properties ¬
        {name:"Big Blue", color:{red:0.0, green:0.0, blue:255.0}}
    set default stroke color of document 1 to {spot:newSpot, tint:50.0}
end tell
```

star

A class used to create a star-shaped path in an Illustrator document. This object is available only in the context of a `make` command, which creates an instance of the `path item` class. This special class allows you to quickly create complex path items. Properties associated with `path items`, such as `fill color` and `note`, can also be specified at the time of creation.

star object properties

Property	Value type	What it is
`center point`	`fixed point`	Write-once. The center point of the `star`. Default: {200.0, 300.0}
`inner radius`	`real`	Write-once. The inner radius of the `star`. Default: 20.0
`point count`	`integer`	Write-once. The number of points on the `star`. Default: 5
`radius`	`real`	Write-once. The radius of the `star's` points. Default: 50.0
`reversed`	`boolean`	Write-once. If `true`, the `star` path is reversed. Default: `false`

star object commands

[make](#)

Create a star

```
-- Make a 16-pointed star
tell application "Adobe Illustrator"
   make new star in document 1 with properties ¬
       {center point:{200.0, 500.0}, inner radius:70, radius:100, point count:16} ¬

end tell
```

story, stories

A contiguous block of text. A story can contain one or more text frames; if more—the multiple text frames are threaded to form a single story.

story object elements

Elements	Refer to by
character	numeric index, range of elements, before/after another element, satisfying a test
insertion point	numeric index, range of elements, before/after another element, satisfying a test
line	numeric index, range of elements, before/after another element, satisfying a test
paragraph	numeric index, range of elements, before/after another element, satisfying a test
text	numeric index, range of elements, before/after another element, satisfying a test
text frame	name, numeric index, range of elements, before/after another element, satisfying a test
word	numeric index, range of elements, before/after another element, satisfying a test

story object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
class	type class	Read-only. The object's class.
container	reference	Read-only. The object's container.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
length	integer	Read-only. The number of characters in the story. Minimum: 0
properties	record	All properties of this object returned as a record.
selection	list of text	Read-only. The selected text.
text range	text	Read-only. The text in the story.

Using stories

```
-- Story
-- Make a new document and two text frames
-- Set the previous frame of the second text frame to text frame 1
-- Add a story to text frame 1, long enough to overflow to text frame 2
-- Count the number of stories
-- Add a new text frame
-- Count the number of stories

tell application "Adobe Illustrator"
    activate
    make new document
    make new rectangle in document 1 with properties {position:{200, 600}, height:30,
width:50}
    make new text frame in document 1 with properties {name:"Text1", kind:area text,
text path:the result}
    make new rectangle in document 1 with properties {position:{300, 550}, height:200,
width:50}
    make new text frame in document 1 with properties {name:"Text2", kind:area text,
text path:the result}
    set previous frame of text frame "Text2" of document 1 to text frame "Text1" of
document 1
    set the contents of text frame "Text1" of document 1 to "This is two text frames
linked together as one story"
    make new rectangle in document 1 with properties {position:{200, 300}, height:30,
width:150}
    make new text frame in document 1 with properties {name:"Text3", kind:area text,
text path:the result}
    set the contents of text frame "Text3" of document 1 to "Each unlinked textFrame
adds a new story"
end tell
```

SVG export options

Options that can be supplied when exporting a document as an SVG file. See the <u>export</u> command in the command reference for additional details.

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

SVG export options object properties

Property	Value type	What it is
compressed	boolean	If `true`, the exported file should be compressed. Default: `false`
coordinate precision	integer	The decimal precision for element coordinate values. Range: 1 to 7 Default: 3

CSS properties	Valid values:
	   entities
	   style attributes
	   style elements
	   presentation attributes
		How should the CCS properties of the document be included in the exported file. Default: `style attributes`

document encoding	Valid values:
	   ASCII
	   UTF8
	   UTF16
		How the text should be encoded in the document. Default: `ASCII`

DTD	Valid values:
	   SVG 1.0
	   SVG 1.1
	   SVG Basic 1.1
	   SVG Tiny 1.1
	   SVG Tiny 1.1 Plus
		The DTD version to which the exported file conforms. Default: `SVG 1.1`

embed auto kerning	boolean	If `true`, SVG automatic kerning is allowed for the file. Default: `false`
embed raster images	boolean	If `true`, the raster images used in the document should be included in the exported file. Default: `false`
embed text on path	boolean	If `true`, the SVG `text-on-path` construct is allowed for the file. Default: `false`

font subsetting	Valid values:
	   none
	   all glyphs
	   glyphs used
	   common english
	   glyphs used plus english
	   common roman
	   glyphs used plus roman
		Specifies which font glyphs should be included in the exported file. Default: `all glyphs`

Property	Value type	What it is
font type	Valid values: CEF font outline font SVG font	The type of font to be included in the exported file.
include file info	boolean	If `true`, the XMP metadata should be included in the exported file. Default: `false`
include variables and datasets	boolean	If `true`, variables and datasets should be included. Default: `false`
optimize for SVG Viewer	boolean	If `true`, the Adobe namespace should be included. Default: `false`
preserve editability	boolean	If `true`, Illustrator editing capabilities should be preserved when exporting the document. Default: `false`
slices	boolean	If `true`, slice data should be preserved in exported document. Default: `false`

Export to SVG

This handler saves all files in a folder as SVG files with linked raster imaged embedded in the exported files. The `class` property is specified in the record to ensure that Illustrator can determine the save option class.

```
-- Opens files from a predefined source folder in Illustrator
-- then exports them to a predefined destination folder in the chosen format
-- fileList is a list of file names in the source folder
-- filePath is the full path to the source folder
-- destFolder is an alias to a folder where the files are to be saved

on SaveFilesAsSVG(fileList, filePath, destFolder)
    set destPath to destFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destPath & fileName & ".svg"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                export current document to file newFilePath as SVG ¬
                    with options {class:SVG export options ¬
                    , embed raster images:true}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsSVG
```

swatch, swatches

A color swatch or list of swatches contained in a document. The swatches correspond to the swatch palette in the Illustrator user interface. Additional swatches can be created either manually by a user or by a script. The swatch can hold all types of color data (such as pattern, gradient, CMYK, RGB, gray, or spot).

swatch object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `swatch`. Always returns `reference`.
class	type class	Read-only. The `swatch` object's class, which is `swatch`.
color	color info	The color information for this `swatch`.
container	object reference	Read-only. A reference to the document that contains this `swatch`.
default type	type class	Read-only. The default type for the `swatch`. Always returns `reference`.
index	integer	Read-only. The position of this `swatch` in the document.
name	Unicode text	The unique name of the `swatch`.
properties	record	All properties of this object returned as a record.

swatch object commands

count
delete
duplicate
exists
make

Create a swatch

```
-- Make a new swatch
tell application "Adobe Illustrator"
    set swatchCount to count of swatches in document 1
    make new swatch in document 1 with properties ¬
        {name:"My Swatch", color:{red:175.0, green:50.0, blue:0.0}}
end tell
```

swatchgroup, swatchgroups

A group of swatches.

swatchgroup object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `swatchgroup` object's value. Always returns `reference`.
class	type class	Read-only. The `swatchgroup`'s class, which is `swatchgroup`.
container	object reference	Read-only. A reference to the object that contains this `swatchgroup`.
default type	type class	Read-only. The default type for the `swatchgroup`.
index	integer	Read-only. The index of this `swatchgroup`.
name	Unicode text	Read-only. The name of the `swatchgroup`. Defaults to `New swatchgroup nnn`, where *n* is an integer, starting at 1 and increasing with each newly created `swatchgroup`.
properties	record	All properties of this object returned as a record.

swatchgroup object commands

add spot
add swatch
count
delete
duplicate
exists
get all swatches
make

symbol, symbols

A symbol or list of symbols. A `symbol` is an art item that is stored in the Symbols palette, and can be reused one or more times in the document without duplicating the art data. Symbols are contained in documents.

symbol object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `symbol` object's value. Always returns `reference`.
class	type class	Read-only. The `symbol`'s class, which is `symbol`.
container	specifier	Read-only. A reference to the object that contains this `symbol`.
default type	type class	Read-only. The default type for the `symbol`.
index	`integer`	Read-only. The index of this `symbol`.
name	Unicode text	Read-only. The name of the `symbol`. Defaults to `New Symbol` *nnn*, where *n* is an integer, starting at 1 and increasing with each newly created symbol.
properties	`record`	All properties of this object returned as a record.
source art	anything	Read-only. The source art is only used when creating a new `symbol`.

symbol object commands

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

Using symbols

```
-- Symbol Items
-- Make a new document
-- Add rectangles, apply different graphic style to each
-- Add delay of at least a second (allow UI to catch up to scripting plug-in)
-- Make a new symbol for each page item, use the page item as the source art

tell application "Adobe Illustrator"
    activate
    make new document
    repeat with i from 1 to (count of graphic styles in document 1)
        round (i / 2 - (round (i / 2) rounding down)) rounding up
        make new rectangle in document 1 with properties ¬
            {position:{100 + (the result * 150), (50 + i * 70)}, height:20, width:20}
        apply graphic style (i) of document 1 to the result
    end repeat
    delay 2
    repeat with i from 1 to (count of graphic styles in document 1)
        make new symbol in document 1 with properties ¬
            {name:("symbol" & i as string), source art:page item i of document 1}
    end repeat
end tell
```

symbol item, symbol items

An instance of a `symbol` in a document. Symbol items are linked to the `symbol` from which they are created and change with any modification of that `symbol`.

symbol item object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
properties	record	All properties of this object returned as a record.
symbol	symbol	The symbol that was used to create this symbol item.

symbol item object commands

[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[move](#)
[rotate](#)
[scale](#)
[transform](#)
[translate](#)

Create symbol items

```
-- Symbol Items
-- Make a new document, add symbol items from symbols panel

tell application "Adobe Illustrator"
    activate
    make new document
    repeat with i from 1 to (count of symbols in document 1)
        round (i / 2 - (round (i / 2) rounding down)) rounding up
        make new symbol item in document 1 with properties ¬
            {symbol:symbol i of document 1, position:{100 + (the result * 150), (50 + i *
70)}}
    end repeat
end tell
```

tab stop info, tab stops

Tab stop information for a paragraph. All tab stops in a paragraph can be retrieved and specified using `tab stops`, which returns a list of `tab stop info` records.

tab stop info object properties

Property	Value type	What it is

`alignment`	Valid values:
	  `left`
	  `center`
	  `right`
	  `decimal`
		The alignment of the tab stop. Default: `left`

`decimal character`	Unicode text	The character to use for decimal tab stops.
`leader`	Unicode text	The leader dot.
`position`	`real`	The position of the tab stop expressed in points. Default: 0.0

Get tab stops

```
-- Return the tab stops of the first paragraph
tell application "Adobe Illustrator"
    set allTabs to tab stops of paragraph 1 of text frame 1 of document 1
    set docRef to make new document
    set textRef to make new text frame in docRef
    set sText to "PositionLeader"
    repeat with i in allTabs
        set curPosition to position of i
        set curLeader to leader of i
        set sText to sText & return & curPosition & "" & curLeader
    end repeat

    set contents of textRef to sText
    set position of textRef to {100.0, 600.0}
end tell
```

tag, tags

A tag or list of tags associated with a specific page item. Tags allows you to assign an unlimited number of key-value pairs to any page item in a document.

tag object properties

Property	Value type	What it is
`best type`	type class	Read-only. The best type for the tag. Always returns `reference`.
`class`	type class	Read-only. The object's class, which is `tag`.
`container`	object reference	Read-only. A reference to the `page item` that contains this `tag`.
`default type`	type class	Read-only. The default type for the tag. Always returns `reference`.
`index`	integer	Read-only. The index of this `tag` in the `page item`.
`name`	Unicode text	The `tag`'s name.
`properties`	record	All properties of this object returned as a record.
`value`	Unicode text	The data stored in this `tag`.

tag object commands

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

Get tags

```
-- Creates then gets tags for the first page item in the document
tell application "Adobe Illustrator"
    make new document
    set newItem to make rectangle in document 1 with properties {name:"rectPath"}
    set myPosition to position of newItem
    set myBoundsString to ((item 1 of myPosition) & "," & (item 2 of myPosition) as
string) & ","
    set myBoundsString to myBoundsString & ((width of newItem) & "," & (height of
newItem) as string)
    set myTag to (make new tag at newItem)
    set name of myTag to "MyNewTag"
    set value of myTag to myBoundsString
    set URL of newItem to "http://www.adobe.com/"
    set tempProp to properties of myTag
    set myTagPropName to name of tempProp
end tell
```

text

Any text in the contents of a text frame. Text can be accessed using the `character`, `insertion point`, `word`, `line`, `paragraph`, and `text` classes. All text is contained within text frames.

text object elements

Elements	Refer to by
`character style`	name, numeric index, range of elements, before/after another element, satisfying a test
`character`	numeric index, range of elements, before/after another element, satisfying a test
`insertion point`	numeric index, range of elements, before/after another element, satisfying a test
`line`	numeric index, range of elements, before/after another element, satisfying a test
`paragraph style`	name, numeric index, range of elements, before/after another element, satisfying a test
`paragraph`	numeric index, range of elements, before/after another element, satisfying a test
`text`	numeric index, range of elements, before/after another element, satisfying a test
`word`	numeric index, range of elements, before/after another element, satisfying a test

text object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value.
character offset	integer	Offset of the first character.
class	type class	Read-only. The object's class.
container	reference	Read-only. The object's container.
contents	Unicode text	The text content.
default type	type class	Read-only. The default type for the object's value.
index	integer	Read-only. The index of this instance of the object.
kerning	integer	Controls the spacing between two characters, in thousandths of an em.
length	integer	The length (in characters). Minimum: 0
properties	record	All properties of this object returned as a record.
selection	list of <u>text</u>	Read-only. The selected text.
story	story	Read-only. The story that contains the text object.

text object commands

Change point size of text

```
-- Change all 12pt text to 18pt
tell application "Adobe Illustrator"
    set textArtItemCount to count text frames of document 1
    -- Loop through all the text frames
    repeat with itemCount from 1 to textArtItemCount
        set textRef to text of text frame itemCount of document 1 ¬
            as reference
        if (size of textRef = 12) then
            set size of textRef to 18
        end if
    end repeat
end tell
```

text font, text fonts

An installed font.

text font object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `text font`.
default type	type class	Read-only. The default type for the object. Always returns `reference`.
family	Unicode text	Read-only. The font's family name.
index	`integer`	Read-only. The index of this object in the art item.
name	Unicode text	The full name of the font.
properties	`record`	All properties of this object returned as a record.
style	Unicode text	Read-only. The font's style name.

text frame, text frames

The basic art item for displaying text. From the user interface, this is text created with the Text tool. There are three types of text art in Illustrator: point text, path text, and area text. The type is specified by the text frame's kind property.

When you create a text frame, you also create a `story` object (see story, stories); however, threading text frames combines the frames into a single story object. To thread frames, use the next frame or previous frame property.

text frame object elements

Element	Refer to by
character	index, before/after, range, test
insertion point	index, before/after, range, test
line	index, before/after, range, test
paragraph	index, before/after, range, test
text	index, before/after, range
word	index, before/after, range, test

text frame object properties

This class inherits all properties from the `page item` class.

Property	Value type	What it is
anchor	list	The position of the anchor point (start of base line for point text).

antialias	Valid values:
	  crisp
	  none
	  sharp
	  strong
		The type of anti-aliasing to use in the text.

column gutter	real	The column gutter in the text frame (area text only).
content variable	anything	The content variable to which this `text frame` is bound. It is not necessary to set the type of the `content variable` before binding. Illustrator automatically sets the type to be the same as the `page item` to which it is bound.
contents	Unicode text	The textual contents of the `text frame`, represented as a string.
column count	integer	The column count in the text frame (area text only).
end T value	real	The end position of text along a path, as a value relative to the path's segments (path text only).

Property	Value type	What it is
flow links horizontally	`boolean`	If `true`, the text flows horizontally first between linked frames.

kind	Valid values:
	`point text`
	`area text`
	`path text`
		The type of text frame.

matrix	`matrix`	Read-only. The transformation matrix of the text frame.
next frame	`text frame`	The linked text frame following this one.
optical alignment	`boolean`	If `true`, the optical alignment is active.
previous frame	`text frame`	The linked text frame preceding this one.
properties	`record`	All properties of this object returned as a record.
row count	`integer`	The row count in the text frame (area text only).
row gutter	`real`	The row gutter in the text frame (area text only).
selection	object reference	Read-only. The selected text.
spacing	`real`	The amount of spacing.
start T value	`real`	The start position of text along a path, as a value relative to the path's segments. **NOTE:** Valid only when kind is `path text`.
story	`story`	Read-only. The story to which the text frame belongs.

text orientation	Valid values:
	`horizontal`
	`vertical`
		The orientation of the text in the frame.

text path	list of path point info	Read-only. The path item associated with the text frame. **NOTE:** Valid only when kind is `area text` or `path text`.
text range	Unicode text	Read-only. The text in the text frame.

text frame object commands

apply character style
apply paragraph style
change case
convert to paths
count
delete
deselect
duplicate
exists
move
rotate
scale
select
transform
translate

Create and manipulate text frames

```
-- Text Frames
-- Make a new document, one text frame of each type: Area, Point, and Path
-- Display the count of text frames
-- Change the contents of each text frame
-- Delete the point text frame
-- Display the count of text frames

tell application "Adobe Illustrator"
    activate
    make new document
    make new rectangle in document 1 with properties ¬
        {position:{100, 700}, height:100, width:100}
    make new text frame in document 1 with properties ¬
        {name:"AreaText", kind:area text, text path:the result, contents:"Text Frame 1"}
¬

    set pathPoint1 to {class:path point info, anchor:{250, 700}}
    set pathPoint2 to {class:path point info, anchor:{350, 550}}
    make new path item in document 1 with properties ¬
        {entire path:{pathPoint1, pathPoint2}}
    make new text frame in document 1 with properties ¬
        {name:"PathText", kind:path text, text path:the result, contents:"Text Frame 2"}
¬

    make new text frame in document 1 with properties ¬
        {name:"PointText", contents:"Text Frame 3"}
    set the position of text frame "PointText" of document 1 to {400, 700}
    set the contents of text frame "AreaText" of document 1 ¬
        to "Area Text is cool"
    set the contents of text frame "PathText" of document 1 ¬
        to "Path Text is cooler"
    set the contents of text frame "PointText" of document 1 ¬
        to "Point Text is not"
    delay 1
    delete text frame "PointText" of document 1
end tell
```

Scale area text frames

```
-- Scale all area text frames to 50% wide
tell application "Adobe Illustrator"
    set textArtItemCount to count text frames in document 1
    set countOTFChanged to 0
    repeat with itemCount from 1 to textArtItemCount
        set textKind to kind of text frame itemCount of document 1
        if (textKind = area text) then
            set curwidth to the width of text frame itemCount of document 1
            set width of text frame itemCount of document 1 to curwidth / 2
            set countOTFChanged to countOTFChanged + 1
        end if
    end repeat
end tell
```

text path item, text path items

A path or list of paths for area or path text. A path consists of path points that define its geometry.

text path item object elements

Element	Refer to by
path point	index, range of elements, before/after another element, satisfying a test

text path item object properties

This object class inherits all properties from the `page item` class.

Property	Value type	What it is
area	real	Read-only. The area of this path in square points. An area may be negative or even 0. The paths winding order is determined by the sign of area. If the area is negative, the path is wound counter-clockwise. Self-intersecting paths may contain sub-areas that cancel each other out. Therefore, it is possible for a path's area to appear as zero even though it has apparent area.
blend mode	Valid values: color blend color burn color dodge darken difference exclusion hard light hue lighten luminosity multiply normal overlay saturation blend screen soft light	The mode to use when compositing this object. An object is considered composited when its opacity is set to less than 100.0 (100%).
clipping	boolean	If `true`, use this path as a clipping path.
closed	boolean	If `true`, this path closed.
container	reference	Read-only. A reference to the art item that contains this path.
editable	boolean	If `true`, this path can be modified.
entire path	list of path point info	All the path item's path points.
evenodd	boolean	If `true`, use the even-odd rule to determine insideness.

Property	Value type	What it is
fill color	<u>color info</u>	The fill color of the path.
fill overprint	boolean	If `true`, the art beneath a filled object should be overprinted.
filled	boolean	If `true`, the path should be filled.
guides	boolean	If `true`, this path is a guide object.
height	real	The height of the path in points. Range: 0.0 to 16348.0
note	Unicode text	The note text assigned to the path.
opacity	real	The object's opacity, expressed as a percentage. Range: 0.0 to 100.0.

polarity	Valid values:
	  `positive`
	  `negative`
		The polarity of the path, used in the creation of compound paths.

position	list	The position (in points) of the top left corner of the item in the format {x, y}. Does not include stroke weight.
resolution	real	The resolution of the path in dots per inch.
selected path points	list of object references	Read-only. All selected path points in the path.

stroke cap	Valid values:
	  `butted`
	  `rounded`
	  `projecting`
		The type of line capping.

stroke color	<u>color info</u>	The stroke color for the path.
stroke dash offset	real	The default distance into the dash pattern at which the pattern should be started
stroke dashes	list of `real` numbers	The lengths for dashes and gaps in dashed lines, starting with the first dash length, followed by the first gap length, and so on. Set to an empty list, `{ }`, for a solid line.

stroke joi	Valid values:
	  `mitered`
	  `rounded`
	  `beveled`
		Type of join for the path.

stroke miter limit	real	When a default stroke join is set to `mitered`, this property specifies when the join will be converted to beveled (squared-off) by default. The default miter limit of 4 means that when the length of the point reaches four times the stroke weight, the join switches from a miter join to a bevel join. Values: 1 to 500. 1 specifies a bevel join.
stroke overprint	boolean	If `true`, the art beneath the stroked object should be overprinted.
stroke width	real	Width of stroke.

Property	Value type	What it is
stroked	`boolean`	If `true`, the path should be stroked.
width	`real`	The width of the text path in points. Range: 0.0 to 16348.0

text path item object commands

[count](#)
[delete](#)
[duplicate](#)
[exists](#)
[move](#)
[rotate](#)
[scale](#)
[transform](#)
[translate](#)

TIFF export options

Options that can be supplied when exporting a document as a TIFF file. See the <u>export</u> command for additional details.

This class contains properties that specify options to be used when exporting a document as a TIFF file. `TIFF export options` can only be supplied in conjunction with the `export` command. It is not possible to get or create a `TIFF export options` object.

TIFF export options object properties

Property	Value type	What it is
antialiasing	boolean	If `true`, the exported image should be anti-aliased. Default: `true`
artboard range	string	If `save multiple artboards` is `true`, this is considered for multi-asset extraction, which specifies the artboard range. An empty string extracts all the artboards. Default: empty string

byte order	Valid values:
	  IBM PC
	  MACINTOSH
		The byte order to use in the new file.

imagecolorspace	Valid values:
	  Gray
	  RGB
	  CMYK
		The color space of the exported file. Default: `RGB`

lzwcompression	boolean	If `true`, use IZW compression in the new file.
resolution	real	Resolution of the exported file in dots per inch (dpi). Range: 72.0 to 2400.0. Default: 150.0
save multiple artboards	boolean	If `true`, all artboards or range of artboards are saved. Default: `false`

tracingobject, tracings

Associates source raster art item with a vector-art plugin group created by tracing. Scripts can initiate tracing using the `trace placed` command for a `placed item` or `raster item`. The resulting `plugin item` object represents the vector art group, and has this object in its `tracing` property.

A script can force the tracing operation by calling the application's `redraw` command. The operation is asynchronous, so a script should call `redraw` after creating the tracingobject, but before accessing its properties or expanding the tracing to convert it to an art item group.

The read-only properties that describe the tracing result have valid values only after the first tracing operation completes. A value of 0 indicates that the operation has not yet been completed.

tracingobject object properties

Property	Value type	What it is
anchor count	`integer`	Read-only. The number of anchors in the tracing result.
area count	`integer`	Read-only. The number of areas in the tracing result.
best type	type class	Read-only. The best type for the object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `text font`.
container	object reference	Read-only. A reference to the object that contains this tracing group.
default type	type class	Read-only. The default type for the object. Always returns `reference`.
image resolution	`real`	Read-only. The resolution of the source image in pixels per inch.
original art	`placed item` or `raster item)`	Read-only. The raster art used to create the associated vector-art plugin group.
path count	`integer`	Read-only. The number of paths in the tracing result.
properties	`record`	All properties of this object returned as a record.
tracing options	`tracing options`	Read-only. The options used to convert the raster artwork to vector art.
used color count	`integer`	Read-only. The number of colors used in the tracing result.

tracingobject object commands

[expand tracing](#)
[release tracing](#)

tracing options, multiple tracing options

A set of options used in converting raster art to vector art by tracing.

tracing options object properties

Property	Value type	What it is
`best type`	type class	Read-only. The best type for the object's value. Always returns `reference`.
`class`	type class	Read-only. The object's class, which is `text font`.
`container`	object reference	Read-only. A reference to the object that contains this tracing group.
`corner angle`	real	The sharpness, in degrees of a turn in the original image that is considered a corner in the tracing result path. Range: 0 to 180
`default type`	type class	Read-only. The default type for the object. Always returns `reference`.
`fills`	boolean	If `true`, trace with fills. At least one of `fills` or `strokes` must be `true`.
`ignore white`	boolean	If `true`, ignores white fill color.
`live paint output`	boolean	If `true`, result is LivePaint art. If `false`, it is classic art. **NOTE:** A script should set this value only in preparation for a subsequent expand operation. Leaving a tracing on the artboard when this property is `true` can lead to unexpected application behavior.
`maximum colors`	integer	The maximum number of colors allowed for automatic palette generation. Used only if `tracing mode` is color or grayscale. Range: 2 to 256
`maximum stroke weight`	real	The maximum stroke weight, when `strokes` is `true`. Range: 0.01 to 100.0
`minimum area`	integer	The smallest feature, in square pixels, that is traced. For example, if it is 4, a feature of 2 pixels wide by 2 pixels high is traced.
`minimum stroke length`	real	The minimum length in pixels of features in the original image that can be stroked, when `strokes` is `true`. Smaller features are omitted. Range: 0.0 to 200.0 Default: 20.0
`output swatches`	boolean	If `true`, named colors (swatches) are generated for each new color created by the tracing result. Used only if `tracing mode` is color or grayscale.

Property	Value type	What it is
palette	string	The name of a color palette to use for tracing. If the empty string, use the automatic palette. Used only if `tracing mode` is color or grayscale.
path fitting	real	The distance between the traced shape and the original pixel shape. Lower values create a tighter path fitting. Higher values create a looser path fitting. Range: 0.0 to 10.0
preprocess blur	real	The amount of blur used during preprocessing. Blurring helps reduce small artifacts and smooth jagged edges in the tracing result. Range: 0.0 to 2.0
preset	string	Read-only. The name of a preset file containing these options.
properties	record	All properties of this object returned as a record.
resample	boolean	If `true`, resample when tracing. (This setting is not captured in a preset file.) Always `true` when the raster source art is placed or linked.
resample resolution	real	The resolution to use when resampling in pixels per inch (ppi). Lower resolution increases the speed of the tracing operation. (This setting is not captured in a preset file.)
strokes	boolean	If `true`, trace with strokes. At least one of `fills` or `strokes` must be `true`. Used only if `tracing mode` is black-and-white.
threshold	integer	The threshold value of black-and-white tracing. All pixels with a grayscale value greater than this are converted to black. Used only if `tracing mode` is black-and-white. Range: 0 to 255

tracing mode	Valid values:
	    `bw tracing mode`
	    `color tracing mode`
	    `gray tracing mode`
		The color mode for tracing.

view raster	Valid values:
	    `view adjusted image`
	    `view no image`
	    `view original image`
	    `view transparent image`
		The view for previews of the raster image. (This setting is not captured in a preset file.)

view vector	Valid values:
	    `view no tracing result`
	    `view outlines`
	    `view outlines tracing`
	    `view tracing result`
		The view for previews of the vector result. (This setting is not captured in a preset file.)

tracing options object commands

[load preset](#)
[store preset](#)

variable, variables

A document-level variable that can be imported or exported.

A variable is a dynamic object used to create data-driven graphics. For an example, see dataset, datasets. Variables are accessed in Illustrator through the Variables palette.

variable object elements

Element	Refer to by
`page item`	name, numeric index, range of elements, before/after another element, satisfying a test

variable object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `variable` object's value. Always returns `reference`.
class	type class	Read-only. The object's class, which is `variable`.
container	object reference	Read-only. A reference to the art item that contains this variable.
default type	type class	Read-only. The default type for the variable. Always returns `reference`.
index	`integer`	Read-only. The index of this `variable` in the art item.

kind	Valid values:
	`graph`
	`image`
	`textual`
	`unknown`
	`visibility`
		The kind of variable.

name	Unicode text	The name of the variable.
properties	`record`	All properties of this object returned as a record.

variable object commands

count
delete
exists
make

view, views

A document view or list of views in an Illustrator document. The `view` object represents a window view onto a document. Scripts cannot create new views, but can modify some properties of existing views, including the center point, screen mode, and zoom.

view object properties

Property	Value type	What it is
best type	type class	Read-only. The best type for the `view` object. Always returns `reference`.
bounds	rect	Read-only. The bounding rectangle of this `view` relative to the current document's bounds
center point	fixed point	The center point of this `view` relative to the current document's bounds
class	type class	Read-only. The object's class, which is `view`.
container	object reference	Read-only. A reference to the document that contains this `view`.
default type	type class	Read-only. The default type for the `view` object. Always returns `reference`.
index	integer	Read-only. The index of the view in the document.
properties	record	All properties of this object returned as a record.

screen mode	Valid values:
	  multiwindow
	  desktop
	  full screen
		The mode of display for this `view`.

zoom	real	The zoom factor of this `view`, where 1.0 is 100%.

view object commands

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

Center a view

```
-- Center the view on the first selected object
tell application "Adobe Illustrator"
    set selectedItems to the selection
    if selectedItems is not {} then
        set firstObject to item 1 of selectedItems
        set newPosition to position of firstObject
        set center point of view 1 of document 1 to newPosition
    end if
end tell
```

Make a view full screen

```
-- Fill the entire screen with the first view
tell application "Adobe Illustrator"
    if (count documents) > 0 then
        set screen mode of view 1 of document 1 to full screen
    end if
end tell
```

word

A string of text in a `text frame` that is separated by whitespace. A document's text can be accessed using the `character`, `insertion point`, `word`, `line`, `paragraph,` and `text` classes. All text is contained within `text frames`.

word object elements

Elements	Refer to by
`character style`	name, numeric index, range of elements, before/after another element, satisfying a test
`character`	numeric index, range of elements, before/after another element, satisfying a test
`insertion point`	numeric index, range of elements, before/after another element, satisfying a test
`line`	numeric index, range of elements, before/after another element, satisfying a test
`paragraph style`	name, numeric index, range of elements, before/after another element, satisfying a test
`paragraph`	name, numeric index, range of elements, before/after another element, satisfying a test
`text`	numeric index, range of elements, before/after another element, satisfying a test
`word`	name, numeric index, range of elements, before/after another element, satisfying a test

word object properties

Property	Value type	What it is
`aki left`	real	The amount of extra space (aki) added to the left side of each glyph in Japanese text (in thousandths of an em).
`aki right`	real	The amount of extra space (aki) added to the right side of each glyph in Japanese text (in thousandths of an em).

`alignment`	Valid values:
	  `bottom`
	  `center`
	  `icf bottom`
	  `icf top`
	  `roman baseline`
	  `top`
		The character alignment type.

Property	Value type	What it is

alternate glyphs	Valid values:
	  `default`
	  `expert`
	  `full width`
	  `half width`
	  `jis78`
	  `jis83`
	  `proportional width`
	  `quarter width`
	  `third width`
	  `traditional`
	  `jis90`
	  `jis04`
		Specifies the type of alternate glyphs.

auto leading	`boolean`	If `true`, use automatic leading.

baseline direction	Valid values:
	  `standard`
	  `Tate Chu Yoko`
	  `vertical rotated`
		Specifies the Japanese text baseline direction.

baseline position	Valid values:
	  `normal`
	  `subscript`
	  `superscript`
		The baseline position of text.

baseline shift	`real`	The amount of shift (in points) of the text baseline.
best type	type class	Read-only. The best type for the object's value.

capitalization	Valid values:
	  `all caps`
	  `all small caps`
	  `normal`
	  `small caps`
		Specifies whether the text is normal, all uppercase, all small caps, or a mix of small caps and lowercase.

character offset	integer	Offset of the first character.
class	type class	Read-only. The object's class.
connection forms	`boolean`	If `true`, use the OpenType connection forms.
container	`reference`	Read-only. The object' s container.
contents	Unicode text	The text content.
contextual ligature	`boolean`	If `true`, use the contextual ligature.
default type	type class	Read-only. The default type for the object's value.
discretionary ligature	`boolean`	If `true`, use the discretionary ligature.

Property	Value type	What it is

`alternate glyphs`	Valid values:
	  `default`
	  `expert`
	  `full width`
	  `half width`
	  `jis78`
	  `jis83`
	  `proportional width`
	  `quarter width`
	  `third width`
	  `traditional`
	  `jis90`
	  `jis04`
		Specifies the type of alternate glyphs.

`auto leading`	`boolean`	If `true`, use automatic leading.

`baseline direction`	Valid values:
	  `standard`
	  `Tate Chu Yoko`
	  `vertical rotated`
		Specifies the Japanese text baseline direction.

`baseline position`	Valid values:
	  `normal`
	  `subscript`
	  `superscript`
		The baseline position of text.

`baseline shift`	`real`	The amount of shift (in points) of the text baseline.
`best type`	`type class`	Read-only. The best type for the object's value.

`capitalization`	Valid values:
	  `all caps`
	  `all small caps`
	  `normal`
	  `small caps`
		Specifies whether the text is normal, all uppercase, all small caps, or a mix of small caps and lowercase.

`character offset`	`integer`	Offset of the first character.
`class`	`type class`	Read-only. The object's class.
`connection forms`	`boolean`	If `true`, use the OpenType connection forms.
`container`	`reference`	Read-only. The object' s container.
`contents`	Unicode text	The text content.
`contextual ligature`	`boolean`	If `true`, use the contextual ligature.
`default type`	`type class`	Read-only. The default type for the object's value.
`discretionary ligature`	`boolean`	If `true`, use the discretionary ligature.

Property	Value type	What it is
figure style	Valid values: default proportional proportional oldstyle tabular tabular oldstyle	Specifies which figure style to use in an OpenType font.
fill color	color info	The color of the text fill.
fractions	boolean	If true, use the OpenType fractions.
horizontal scale	real	The character horizontal scaling factor expressed as a percentage (100 = 100%).
index	integer	Read-only. The index of this instance of the object.
italics	boolean	If true, the Japanese font supports italics.
kerning	integer	Controls the spacing between two characters, in thousandths of the em space.
kerning method	Valid values: auto none optical	The type of automatic kerning method to use.

Property	Value type	What it is

language	Valid values:
	  `Bokmal Norwegian`
	  `Brazillian Portuguese`
	  `Bulgarian`
	  `Canadian French`
	  `Catalan`
	  `Chinese`
	  `Czech`
	  `Danish`
	  `Dutch`
	  `Dutch 2005 Reform`
	  `English`
	  `Finnish`
	  `German 2006 Reform`
	  `Greek`
	  `Hungarian`
	  `Icelandic`
	  `Italian`
	  `Japanese`
	  `Nynorsk Norwegian`
	  `old German`
	  `Polish`
	  `Romanian`
	  `Russian`
	  `Spanish`
	  `Serbian`
	  `standard French`
	  `standard German`
	  `standard Portuguese`
	  `Swedish`
	  `Swiss German`
	  `Swiss German 2006 Reform`
	  `Turkish`
	  `UK English`
	  `Ukranian`
		The language.

leading	`real`	The amount of space between two lines of text, in points.
length	`integer`	Read-only. The length (in characters). Minimum: 0
ligature	`boolean`	If `true`, use the ligature.
no break	`boolean`	If `true`, no line break is allowed in this word.

OpenType position	Valid values:
	  `default`
	  `denominator`
	  `numerator`
	  `subscript`
	  `superscript`
		The OpenType baseline position.

ordinals	`boolean`	If `true`, use the OpenType ordinals.
ornaments	`boolean`	If `true`, use the OpenType ornaments.

Property	Value type	What it is
overprint fill	boolean	If true, overprint the fill of the text.
overprint stroke	boolean	If true, overprinting of the stroke of the text is allowed.
properties	record	All properties of this object returned as a record.
proportional metrics	boolean	If true, the Japanese OpenType supports proportional fonts.
rotation	real	The character rotation angle in degrees.
selection	list of text	Read-only. The selected text.
size	real	The font size in points.
story	story	Read-only. The story that contains the object.
strike through	boolean	If true, characters use strike-through style.
stroke color	color info	The color of the text stroke.
stroke weight	real	Line width of stroke.
stylistic alternates	boolean	If true, use OpenType stylistic alternates.
swash	boolean	If true, use the OpenType swash character.
TCY horizontal	integer	The Tate-Chu-Yoko horizontal adjustment in points.
TCY vertical	integer	The Tate-Chu-Yoko vertical adjustment in points.
text font	text font	The text font.
titling	boolean	If true, use the OpenType titling alternates.
tracking	integer	The tracking or range kerning amount in thousandths of an em.
Tsume	real	The percentage of space reduction around a Japanese character.
underline	boolean	If true, characters use underline style.
vertical scale	real	Character vertical scaling factor. 100 = 100%
warichu characters after break	integer	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.

Property	Value type	What it is
warichu characters before break	integer	How the characters in Wari-Chu text (an inset note in Japanese text) are divided into two or more lines.
warichu enabled	boolean	If true, Wari-Chu is enabled.
warichu gap	integer	The Wari-Chu line gap.
warichu justification	Valid values: auto justify center full justify last line center full justify full justify last line left full justify last line right left right	The Wari-Chu justification.
warichu lines	integer	The number of Wari-Chu (multiple text lines fit into a space meant for one) lines.
warichu scale	real	The Wari-Chu scale.

word object commands

apply character style
apply paragraph style
change case
count
delete
deselect
duplicate
exists
make
select

Finding specific words

This example demonstrates how to use the matching abilities of the `whose` clause in conjunction with word properties to modify words that match a specific string.

```
-- Change the color of every occurrence of a specific
-- word in all text frames

set searchString to "the"

tell application "Adobe Illustrator"
    set textArtItemCount to (count text frames in document 1)
    if (textArtItemCount > 0) then

        repeat with itemCounter from 1 to textArtItemCount
            if (((contents of text frame itemCounter of document 1) as string) ¬
                contains searchString) then
                set fill color of (words of text frame itemCounter of document 1 ¬
                    whose contents contains searchString) to {red:255, green:0, blue:0}
            end if
        end repeat
    end if
end tell
```

2   AppleScript Commands

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

The commands supported by each object, with links to the detailed descriptions here, are listed in Chapter 1, "AppleScript Objects."

Overview

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

The following information is given for each command:

Column heading	What it means
Parameters	Constants, keywords, and values needed by the command.
	▶ Variable values to be supplied are shown in bold.
	▶ Literal terms and constants are shown in plain type.
	▶ Items surrounded by brackets `[ ]` are optional.
What it is	An explanation of the parameters.
Objects supported	Which objects support the command and/or which objects the command can operate upon. The `document` object, for example, supports the command `close`, but not the command `quit`.
Returns	Many commands return values (text, numbers, lists, and object references). This column shows you what kind of reference you can expect the command to return (if any).

activate

Makes an application active; that is, makes Illustrator the front-most application.

Parameters	What it is	Objects supported	Returns
none		application	nothing

Notes Illustrator must be the frontmost application when executing any command that deals with the clipboard. Use this command to ensure this. See the clipboard commands for examples.

add document

Creates a new document from a preset template.

Parameters	What it is	Objects supported	Returns
using **startup preset**	The document template.	document	document
with **preset settings**	The preset document settings.		

add spot

Adds a spot swatch to the swatch group.

Parameters	What it is	Objects supported	Returns
spot **spot**	The spot swatch to be added.	swatch	nothing

add swatch

Adds a swatch to the swatch group.

Parameters	What it is	Objects supported	Returns
swatch **swatch**	The swatch to be added.	swatch	nothing

apply

Applies a brush or graphic style to one or more page items.

Parameters	What it is	Objects supported	Returns
`object reference`	The brush or graphic style to apply to the referenced objector objects.	`graphic style` `brush`	nothing

to `anything`	The page item or items to which to apply abrush or graphic style.	`compound path item`
		`group item`
		`mesh item`
		`non native item`
		`page item`
		`path item`
		`placed item`
		`plugin item`
		`raster item`
		`text frame`	

Notes   Use `apply` to affect one or more page items by applying an existing brush or graphic style. Brushes and graphic styles can be created in the user interface, but not from a script.

Apply an art style

```
-- Draws an ellipse in the center of the document
-- and applies a graphic style to it
tell application "Adobe Illustrator"
    make new document with properties {color space:CMYK}
    set docWidth to (width of document 1) / 2
    set docHeight to (height of document 1) / 2
    set pathItemRef to make new ellipse in document 1 with properties ¬
        {bounds:{docWidth - 50, docHeight + 50, docWidth + 50, docHeight - 50}}
    apply graphic style 2 of document 1 to pathItemRef
end tell
```

apply character style

Applies a character style to a specified text object(s).

Parameters	What it is	Objects supported	Returns
`character style`	The character style object or objects to be operated upon.	`character style`	nothing
`to anything`	The text object or objects to which to apply the style.	text	
`[clearing overrides boolean]`	Whether to clear any text attributes before apply the style. Default: `false`		

apply paragraph style

Applies the paragraph style to text object(s).

Parameters	What it is	Objects supported	Returns
paragraph style	The paragraph style object or objects to be operated upon.	paragraph style	nothing
to **anything**	The text object or objects to which to apply the style.	text	
[clearing overrides **boolean**]	If true, text attributes are cleared before apply the style. Default: false		

bring in perspective

Places art object(s) in a perspective grid at a specified position and grid plane.

Parameters	What it is	Objects supported	Returns
`page item`	The page item object or objects to be operated upon.	`page item`	nothing
`position x` **real**	Horizontal coordinate at which to place art on grid plane.		
`position y` **real**	Vertical coordinate at which to place art on grid plane.		
`[perspective grid plane floorplane/leftplane/ noplane/rightplane]`	Type of perspective grid plane.		

capture

Captures the current document window to the target TIFF image file.

Parameters	What it is	Objects supported	Returns
`to` **`file specification`**	The TIFF file to which the captured image should be written.	`document`	nothing
`size` **`point`**	The size to make the window before capture.		

change case

Changes the capitalization of the selected text.

Parameters	What it is	Objects supported	Returns
text	The text object or objects to be operated upon.	<u>text</u>	nothing

to lower case/
  sentence case/
  title case/
  upper case
	The type of case.		

close

Closes a document.

Parameters	What it is	Objects supported	Returns
`document`	The document to close.	`document`	nothing
`[saving yes/no/ask]`	Whether to save the document before closing.		

Close a document

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

colorize

Colorizes a raster item.

Parameters	What it is	Objects supported	Returns
object reference	The raster item to colorize.	`raster item`	nothing
`raster color` **color info reference**	The color to use when coloring the TIFF image.	[CMYK color info](#) [gradient color info](#) [gray color info](#) [pattern color info](#) [RGB color info](#) [spot color info](#)	

concatenate matrix

Concatenates two transformation matrices to form a single resulting matrix.

Parameters	What it is	Objects supported	Returns
matrix	The first matrix.	matrix	matrix
with **matrix**	The second matrix.	matrix	

Concatenate matrices

```
-- This script concatenates 2 matrices
tell application "Adobe Illustrator"
    set someMatrix to get identity matrix
    set anotherMatrix to get rotation matrix angle 30.0
    set newMatrix to concatenate matrix someMatrix with anotherMatrix
end tell
```

concatenate rotation matrix

Concatenates a rotation angle together with a matrix and returns the resulting matrix.

Parameters	What it is	Objects supported	Returns
`matrix`	The matrix.	`matrix`	`matrix`
`angle` **`real`**	Rotation angle in degrees.		

Concatenate rotation matrix

```
-- This script adds a 45 degree rotation to an existing matrix
tell application "Adobe Illustrator"
    set someMatrix to get identity matrix
    set newMatrix to concatenate rotation matrix someMatrix angle 45.0
end tell
```

concatenate scale matrix

Concatenates a horizontal and/or vertical scaling with a matrix to form a new, rescaled matrix.

Parameters	What it is	Objects supported	Returns
`matrix`	The matrix.	`matrix`	`matrix`
`[horizontal scale real]`	Horizontal scaling factor, 100.0 is 100%. Default: 100.0		
`[vertical scale real]`	Vertical scaling factor, 100.0 is 100%. Default: 100.0		

Concatenate scale matrix

```
-- This script combines a 75% horizontal and 25% vertical scaling with an existing
matrix
tell application "Adobe Illustrator"
    set someMatrix to get identity matrix
    set newMatrix to concatenate scale matrix someMatrix ¬
        horizontal scale 75 vertical scale 25.0
end tell
```

concatenate translation matrix

Concatenates a positional translation factor (specified by a horizontal and/or vertical offset) with a matrix to form a new, repositioned matrix.

Parameters	What it is	Objects supported	Returns
matrix	The matrix.	matrix	matrix
[delta x **real**]	Horizontal translation offset. Default: 0.0		
[delta y **real**]	Vertical translation offset. Default: 0.0		

Concatenate translation matrix

```
--This script combines a 25 point horizontal offset with an existing matrix
tell application "Adobe Illustrator"
    set someMatrix to get identity matrix
    set newMatrix to concatenate translation matrix someMatrix delta x 25.0
end tell
```

convert

Converts the legacy text item to a text frame and deletes the original legacy text item.

Parameters	What it is	Objects supported	Returns
legacy text item	The `legacy text item` object or objects to be operated upon.	legacy text item	group item

convert coordinate

Converts the coordinates of a single point from one coordinate system to another. Returns the modified coordinate list.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated upon.	document	list
coordinate list	The coordinates of the point to be converted.		

source	The current coordinate system. Valid values:
	`artboard coordinate system`
	`document coordinate system`		

destination	The system to which to convert. Valid values:
	`artboard coordinate system`
	`document coordinate system`		

convert sample color

Converts a sample-component color from one color space to another.

Parameters	What it is	Objects supported	Returns
source color space **image color space**	The source color space.	application	array of ColorComponents
source color **color components**	The color to convert, a color sample object. The first location should contain alpha if source has alpha is true.		An array of color components(e.g. R,G and B value) which constitute the color. First location of returned array will contain alpha value if destination-has-alpha is true.
destination color space **image color space**	The destination color space. The first location should contain alpha if dest has alpha is true.		

color convert purpose **color conversion purpose**	The purpose of conversion. Valid values:
	   default purpose
	   dummy purpose option
	   export purpose
	   preview purpose		

[source has alpha **boolean**]	True if the alpha channel is present in the source color. Default: false		
[destination has alpha **boolean**]	True if the alpha channel is present in the destination color. Default: false		

convert to paths

Converts the specified text to path items.

Parameters	What it is	Objects supported	Returns
`text frame`	The `text frame` object or objects to be operated upon.	`text frame`	`group item` or null

Create outlines from text

```
--This script converts all text art to path art
tell application "Adobe Illustrator"
    convert to paths (every text frame of document 1)
end tell
```

copy

Copies the selection in the current document to the clipboard.

Parameters	What it is	Objects supported	Returns
none		compound path item group item mesh item non native item path item placed item plugin item raster item text frame	nothing

Notes Commands that manipulate the clipboard (`cut`, `copy`, and `paste`) require that Illustrator be the frontmost application during these operations. Use `activate` to bring Illustrator to the front before executing the `copy` command. No error is returned if there is no selection to copy. If the application is not frontmost, an error is returned.

Copy selected objects

```
--This script copies the selected objects (if any) to the clipboard
tell application "Adobe Illustrator"
    activate
    copy
end tell
```

count

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

Parameters	What it is	Objects supported	Returns
count **reference**	The object or list of objects whose elements are to be counted.	graphic style brush character	integer
[each **type class**]	The class of the objects to count.	compound path item document gradient gradient stop group item insertion point layer line mesh item non native item page item paragraph path item path point pattern placed item plugin item raster item spot tag text frame view word	
[whose **property** is **value**]	A condition that objects must meet to be counted.		

Notes   With the optional each/every term, use the singular form for the object type to be counted; for example, brush rather than brushes. Otherwise, you can use the singular or plural form.

Count filled path items in a document

```
-- This script stores the total path items in pathCount and
-- the total filled path items in numberFilled
tell application "Adobe Illustrator"
    set pathCount to count every path item of document 1
    set numberFilled to ¬
        count (path items of document 1 whose filled is true)
end tell
```

cut

Cuts the current selection from the current document and places it in the clipboard.

Parameters	What it is	Objects supported	Returns

none	nothing	`compound path item`
		`group item`
		`mesh item`
		`non native item`
		`path item`
		`path point`
		`placed item`
		`plugin item`
		`raster item`
		`text`
		`text frame`
			nothing

Notes   Commands that manipulate the clipboard (`cut`, `copy`, and `paste`) require that Illustrator be the frontmost application. Use `activate` to bring Illustrator to the front before executing the `cut` command. No error is returned if there is no selection to cut. If the application is not frontmost, an error is returned.

Cut selected objects to the clipboard

```
--This script cuts the selected objects (if any)
tell application "Adobe Illustrator"
    activate
    cut
end tell
```

delete

Removes one or more elements from a container, or deletes one or more objects.

Parameters	What it is	Objects supported	Returns
`object reference`	Contained object or objects to delete or remove. You cannot remove the last artboard in a document.	artboard compound path item gradient gradient stop group item layer mesh item non native item page item point pattern placed item plugin item raster item spot swatch tag text text frame text path	nothing
[of `object reference`]	Container object. If supplied, removes the specified object or objects from this container. If not supplied, deletes the specified object or objects.	document group layer compound path item path item story	

Delete a layer

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

delete preference

Removes the application preference key. See "Preference accessor guidelines" on page 74.

Parameters	What it is	Objects supported	Returns
`Illustrator preferences`	The Illustrator preferences object or objects to delete.		nothing
`key`	The preference key		

deselect

Deselects a text range.

Parameters	What it is	Objects supported	Returns
text	The `text` object or objects to be deselected.	text	nothing

display

Displays the dynamic data that has been captured in a `dataset` object.

.

Parameters	What it is	Objects supported	Returns
dataset	The `dataset` object or objects to be displayed.	`dataset`	`boolean`

do javascript

Executes a JavaScript script and returns the result of execution.

Parameters	What it is	Objects supported	Returns
`javascript`	The JavaScript code to execute.	N/A	Unicode text
`[with arguments list of anything]`	A list of suitable arguments to pass to the Javascript routine.		
`[show debugger before running/ never/ on runtime error]`	When a debugger should be shown. Default: `never`		

do script

Plays an action from the Actions palette.

Parameters	What it is	Objects supported	Returns
Unicode text	The name of the action to play. Case-sensitive.	N/A	nothing
from **Unicode text**	The name of the Action Set containing the action. Case-sensitive.		
[dialogs **boolean**]	If `true`, dialog boxes should be associated with the action presented to the user. Default: `true`		

Notes   If the action is selected in the Actions palette in the Illustrator user interface, this command returns an error.

Execute an action

```
-- This script executes an action in the default set without displaying any dialogs
tell application "Adobe Illustrator"
    do script "Opacity 60 (selection)" from "Default Actions" without dialogs
end tell
```

duplicate

Duplicates an object or objects.

Parameters	What it is	Objects supported	Returns
`object reference`	The object or objects to duplicate	all objects *except*:	object reference or list (of object references)
`[to location reference]`	The location for the new object or objects	application mesh item plugin items	
`[with properties record]`	New values for specified properties of the new object or objects		

Notes   You can duplicate page items from one document to another. This is equivalent to setting the selection, performing a cut or copy, bringing another document to the front, and then pasting. When duplicating objects from one document to another, you must specify the location reference.

Duplicate to another document

```
-- Duplicate the first page item in document 1 to document 2
tell application "Adobe Illustrator"
    set pageItemRef to duplicate page item 1 of document 1 to beginning of document 2
end tell
```

equal matrices

Compares two matrices for equality.

Parameters	What it is	Objects supported	Returns
matrix	The first matrix for the comparison.	matrix	boolean
with **matrix**	The second matrix for the comparison.		

Compare matrices

```
-- This script compares 2 matrices and beeps if they are equal
tell application "Adobe Illustrator"
    set someMatrix to get identity matrix
    set anotherMatrix to get identity matrix
    if (equal matrices someMatrix with anotherMatrix) then beep
end tell
```

embed

Embeds linked art in a document. Applied to a `placed item`, it converts the art to art item objects as needed and deletes the `placed item` object.

Parameters	What it is	Objects supported	Returns
object reference	The `placed item` to embed.	placed item	nothing

exists

Determines whether an object exists.

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

Check if a document exists

```
-- Check if a document exists and beep twice
tell application "Adobe Illustrator"
    if exists document 1 then beep 2
end tell
```

expand tracing

Converts the vector art associated with a `tracingobject` into a new group item. The new `group item` object replaces the `plugin item` object in the document. Deletes this object and its associated `plugin item` object. Any group-level attributes that were applied to the plugin item are applied to the top level of the new group item.

Parameters	What it is	Objects supported	Returns
tracingobject	The `tracingobject` object to operate on.	`tracingobject`	`group item` object reference
[with viewed]	By default the new group contains only the tracing result (the filled or stroked paths). If `with viewed` is specified, the new group retains additional information that was specified for the viewing mode, such as outlines and overlays.		

export

Exports the specified document to a specified file type.

Parameters	What it is	Objects supported	Returns
`object reference`	The document to export.	document	nothing
to `file specification`	The file to export to, specified as a string containing the full file path or an alias.		
as JPEG/Photoshop/ SVG/PNG8/PNG24/ GIF/AutoCAD/ Flash/TIFF	The file type to which to export the document.		
[with options] `object reference`	The export options for the specified file type.		

Export a document to JPEG

```
-- This script exports the current document as JPEG to the
-- destinationFolder passed in as a parameter, destinationFolder
-- is set by the framework this fragment is tested in
on exportFile(destinationFolder)
    set destinationPath to destinationFolder as string
    set newFilePath to destinationPath & "Sample.jpg"
    tell application "Adobe Illustrator"
        export current document to newFilePath as JPEG with options ¬
            {class:JPEG export options, quality:60}
    end tell
end exportFile
```

export PDF preset

Exports PDF presets for a document and saves them to a file.

Parameters	What it is	Objects supported	Returns
`document`	The `document` object or objects to be operated on.	document	nothing
`to` **`file specification`**	The file to export to, specified as a string containing the full file path or an alias.		

export perspective grid preset

Saves all perspective grid presets to a file.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated upon.	`document`	nothing
to **file specification**	The file to export to.		

export print preset

Exports Illustrator print presets for a document to a file.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated on.	document	nothing
`to` **file specification**	The file to export to, specified as a string containing the full file path or an alias.		

export variables

Saves datasets containing variables and their associated dynamic data into an XML library.

Parameters	What it is	Objects supported	Returns
`document`	The `document` object or objects to be operated on.	`document`	nothing
`to file specification`	The file to export to, specified as a string containing the full file path or an alias.		

fit artboard to selected art

Resizes the artboard to the bounds of the currently selected art.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated on.	document	boolean
index	The index of the artboard to update.		

get

Gets data from an object.

Parameters	What it is	Objects supported	Returns
object reference *or* **property**	The object or property to get a reference to or data from.	Any object	The property value or object reference as the specified type.
[as **class** *or* list (of **classes**)]	The type of data to retrieve.		

Notes This standard AppleScript command is included because it illustrates AppleScript's ability to coerce values from one value type to another. You do not need to use get to assign values to variables.

Using the get command

```
-- This script gets the contents of a text frame both as a string and as a reference
tell application "Adobe Illustrator"
    set textString to contents of text frame 1 of document 1
    set textRef to text of text frame 1 of document 1 as reference
end tell
```

get all swatches

Gets a list of all swatches in the swatch group.

Parameters	What it is	Objects supported	Returns
none	Nothing	`swatchgroup`	`swatchlist` (list of selected swatches)

get boolean preference

Gets the value of the application preference key as boolean. See "Preference accessor guidelines" on page 74.

Parameters	What it is	Objects supported	Returns
<u>Illustrator preferences</u>	The `Illustrator preferences` object or objects to be operated on.	<u>Illustrator preferences</u>	The value of the application preference key as boolean.
`key` **as Unicode text**	The type of data to retrieve.		

get identity matrix

Returns an identity matrix.

Parameters	What it is	Objects supported	Returns
none	nothing	`matrix`	`matrix`

Notes   The identity matrix is a transformation matrix that causes no transformation. Use it to get a base matrix to use with the matrix concatenation commands.

Using an identity matrix

```
-- This script gets the identity matrix,
-- combines with rotation and scale and applies to an object
tell application "Adobe Illustrator"
    set transformMatrix to get identity matrix
    set transformMatrix to concatenate scale matrix ¬
        transformMatrix horizontal scale 60
    set transformMatrix to concatenate rotation matrix ¬
        transformMatrix angle 45.0
    transform page item 1 of document 1 using transformMatrix
end tell
```

get internal color

Gets the internal color of a spot.

Parameters	What it is	Objects supported	Returns
none	nothing	spot	color components

get integer preference

Gets the value of the application preference key as an integer. See "Preference accessor guidelines" on page 74.

Parameters	What it is	Objects supported	Returns
Illustrator preferences	The Illustrator preferences object or objects to be operated upon.	Illustrator preferences	The value of the application preference key as an integer.
key **as Unicode text**	The type of data to retrieve.		

get perspective active plane

Retrieves the active plane of the active perspective grid of the document.

Parameters	What it is	Objects supported	Returns
document	The document object or objects to be operated upon.	document	floorplane leftplane noplane rightplane

get PPD info

Gets detailed file information for a specified PPD file.

Parameters	What it is	Objects supported	Returns
name **Unicode text**	The model name of the PPD file	application	PPD properties

get preset file of

Returns the full path to the application's default document profile for the specified preset type.

Parameters	What it is	Objects supported	Returns

`preset type`	The name of the preset type. Valid values:
	`basic CMYK document`
	`basic RGB document`
	`print document preset`
	`mobile document preset`
	`video document preset`
	`web document preset`
		application	`file specification`

get preset settings

Retrieves the tracing-option settings from the template with a given preset name.

Parameters	What it is	Objects supported	Returns
preset **Unicode text**	The name of the preset	application	document preset

get real preference

Gets the value of the application preference key as a real number. See <u>"Preference accessor guidelines" on page 74</u>.

Parameters	What it is	Objects supported	Returns
<u>Illustrator preferences</u>	The Illustrator preferences object or objects to be operated upon.	<u>Illustrator preferences</u>	The value of the application preference key as a real number.
key **as Unicode text**	The type of data to retrieve.		

get rotation matrix

Returns a rotation matrix based on a specified rotation angle.

Parameters	What it is	Objects supported	Returns
[angle **real**]	The rotation angle in degrees. Default is 0.0, which returns the standard identity matrix.	matrix	matrix

Notes   Requires a value in degrees. 30 rotates the object 30 degrees counterclockwise; -30 rotates the object 30 degrees clockwise.

Get a rotation matrix

```
-- Get a 30-degree rotation matrix
tell application "Adobe Illustrator"
    set rotateMatrix to get rotation matrix angle 30.0
end tell
```

get scale matrix

Returns a scale matrix based on specified horizontal and vertical scaling factor.

Parameters	What it is	Objects supported	Returns
[horizontal scale **real**]	The horizontal scaling factor as a percentage. Default is 100.0, which is 100%	matrix	matrix
[vertical scale **real**]	The vertical scaling factor. Default is 100.0, which is 100%		

Notes If no parameters are supplied, returns the standard identity matrix.

Requires a value in percentage. 60 scales the object to 60% of its original size; 200 doubles the objects bounds.

Get a scale matrix

```
-- This script gets a scale matrix
tell application "Adobe Illustrator"
    set scaleMatrix to get scale matrix horizontal scale 100.0 vertical scale 50.0
end tell
```

get scriptable help group

Gets the scriptable help group object that represents the search widget in the app bar.

Parameters	What it is	Objects supported	Returns
		<u>application</u>	`variant` (app bar help group live object)

get selected

Gets the selected swatches in a document.

Parameters	What it is	Objects supported	Returns
		swatches	swatchlist (list of selected swatches)

get string preference

Gets the value of the application preference key as string type. See "Preference accessor guidelines" on page 74.

Parameters	What it is	Objects supported	Returns
Illustrator preferences	The Illustrator preferences object or objects to be operated upon.	Illustrator preferences	The value of the application preference key as string type.
key **as Unicode text**	The type of data to retrieve.		

get translation matrix

Returns a translation matrix based on a single movement with horizontal and vertical offsets.

Parameters	What it is	Objects supported	Returns
[delta x **real**]	The horizontal offset. Default: 0.0	matrix	matrix
[delta y **real**]	The vertical offset. Default: 0.0		

Notes   If no parameters are supplied, returns the standard identity matrix.

Requires a value in points. {100,200} moves the object 100 pt. to the right and 200 pt. up; a minus (-) before each number moves the object left and down.

Get a translation matrix

```
-- This script gets a translation matrix
tell application "Adobe Illustrator"
    set translateMatrix to get translation matrix delta x 10.0 delta y 100.0
end tell
```

hide perspective grid

Hides the current active perspective grid for the document, if one is visible.

Parameters	What it is	Objects supported	Returns
`document`	The document object or objects to be operated upon.	document	boolean

image capture

Captures the artwork content within the clipping boundaries in this document as a raster image, and writes the image data to a specified file.

Parameters	What it is	Objects supported	Returns
`to` **`file specification`**	The file to which the captured image should be written.	`document`	nothing
`[inside `**`rect`**`]`	The rectangular region of the artwork for image capture. If omitted, captures the entire artwork.		
`[with options `**`image capture options`**`]`	The object describing the image-capture options.		

import character styles

Loads character styles from a file.

Parameters	What it is	Objects supported	Returns
`document`	The `document` object or objects to be operated on.	`document`	nothing
`from` **`file specification`**	File from which to import.		

import paragraph styles

Loads paragraph styles from a file.

Parameters	What it is	Objects supported	Returns
`document`	The `document` object or objects to be operated on.	`document`	nothing
from **file specification**	File from which to import.		

import PDF preset

Loads all PDF presets from a file.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated on.	`document`	nothing
from **file specification**	File from which to import.		
[replacing preset **boolean**]	Whether existing editable presets should be replaced. Default: `false`		

import perspective grid preset

Loads a specified perspective grid preset, or all presets, from a specified file.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated upon.	`document`	nothing
from **file specification**	File to import from.		
[perspective preset **text**]	Name of a perspective grid preset. If not supplied, loads all presets from the file.		

import print preset

Loads a print preset from a file.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to be operated on.	document	nothing
`print preset` **Unicode text**	The name of the print preset to import.		
`from` **file specification**	The file to import from, specified as a string containing the full file path or an alias.		

import variables

Loads a library from a file that contains datasets, variables, and the associated dynamic data. The imported data overwrites any existing variables and datasets.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects into which to import variables	`document`	nothing
from **file specification**	The file from which to import variables, specified as a string containing the full file path or an alias.		

invert matrix

Returns an inverted matrix.

Parameters	What it is	Objects supported	Returns
matrix	The matrix to invert.	matrix	matrix

Notes   A singular matrix cannot be inverted. Use the `singular matrix` command to test if a matrix is singular.

Invert a matrix

```
-- This script gets the inverse matrix of a 50% vertical scale matrix
-- When applied, the inverse matrix scales the object 200% vertically
tell application "Adobe Illustrator"
    set transformMatrix to get scale matrix vertical scale 50.0
    set transformMatrix to invert matrix transformMatrix
    transform page item 1 of document 1 using transformMatrix
end tell
```

launch

Launches Illustrator.

Parameters	What it is	Objects supported	Returns
none		application	nothing

load color settings

Loads color settings from specified file, or, if file is empty, turns color management off.

Parameters	What it is	Objects supported	Returns
from **file specification**	The color-settings file	[application](#)	nothing

load preset

Loads a set of preset tracing options from a file into a `tracing options` object.

Parameters	What it is	Objects supported	Returns
tracing options	The `tracing options` object to operate on.	`tracing options`	`boolean`
presetname **Unicode text**	The preset name, as found in the `application` `tracing presets` list.		

make

Creates a new object and returns a reference to newly created object. To place new art in a document, use this command to create a `placed item`, then use the `embed` command on the resulting `placed item` object to convert it to embedded art items.

Parameters	What it is	Objects supported	Returns

`new` **type class**	The class of object to create. The term `new` is optional.	all objects *except*:
		<u>application</u>
		`mesh item`
		`plugin item`
			object reference

`at` **location reference**	Location at which to insert new object.		
`[with properties` **record**`]`	Any property of the object you wish to set at creation.		
`[with data` **anything**`]`	Any data needed for creation that is not a property.		

Create layers

An open document must exist before this script is executed.

```
-- Make 2 layers, one at the top and one at the bottom
-- demonstrating the power of location references like beginning and end
tell application "Adobe Illustrator"
    set topLayer to make new layer ¬
        at beginning of document 1 with properties {name:"Top Layer"}
    set bottomLayer to make new layer ¬
        at end of document 1 with properties {name:"Bottom Layer"}
end tell
```

merge

Merges this style into the current style(s) of the specified items.

Parameters	What it is	Objects supported	Returns
`graphic style`	The graphic style to be merged.	`compound path item` `group item`	The merged style

`graphic style` to `anything`	The object or objects to merge the style into.	`mesh item`
		`non native item`
		`page item`
		`path item`
		`placed item`
		`plugin item`
		`raster item`
		`text frame`	

move

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

Parameters	What it is	Objects supported	Returns
`object reference`	Object or objects to move.	`compound path item` `group item` `layer` `mesh item` `non native item` `page item` `path item` `placed item` `plugin item` `raster item` `text frame`	object reference or list (of object references)
`to location reference`	New location of the object or objects.		

Notes   Objects cannot be moved between documents.

Move objects to a layer

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

Move layers

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

open

Opens one or more specified documents.

Parameters	What it is	Objects supported	Returns
`file specification`	The file to be opened.	N/A	nothing
`[forcing RGB/CMYK]`	Pre-Illustrator 9 files only. Opens the document using the specified color space, converting if necessary. If not supplied, and the document contains both color spaces, displays a dialog for the user to choose one.		
`[dialogs boolean]`	If `true`, show warning and error dialogs when opening the file or files. Default: `true`		
`[with options anything]`	Options for opening a particular type of file.		

Open a PDF file

```
-- This function opens the file passed as
-- a file reference parameter, fileToOpen is
-- a reference to a PDF file and needs to
-- be set up before calling this function
on openFile(fileToOpen)
    tell application "Adobe Illustrator"
        set user interaction level to never interact
        open POSIX file fileToOpen as alias without dialogs
    end tell
end openFile
```

paste

Pastes the clipboard contents into the current layer of the current document.

Parameters	What it is	Objects supported	Returns
none		compound path item group item mesh item non native item path item path point placed item plugin item raster item [text](#) text frame	nothing

Notes   Commands that manipulate the clipboard (`cut`, `copy`, and `paste`) require that Illustrator be the frontmost application. Use `activate` to bring Illustrator to the front before executing the `paste` command. No error is returned if there is no selection to paste. If the application is not frontmost, an error is returned.

Paste from the clipboard

```
-- Paste the contents of the clipboard into the current document
tell application "Adobe Illustrator"
    activate
    paste
end tell
```

print

Prints one or more documents or files.

Parameters	What it is	Objects supported	Returns
anything	Document(s) or file(s) to be printed.	document	nothing
[options **print options**]	A print options object.	print options	

Print a document

```
-- Print the current document without displaying a dialog
tell application "Adobe Illustrator"
    print document 1 without dialog
end tell
```

Print with options

```
-- Make new document. add symbol items
-- Set job options, color management options, coordinate options, flattening options
-- Print the document using these options
tell application "Adobe Illustrator"
    activate
    make new document
    repeat with i from 1 to (count of symbols in document 1)
        round (i / 2 - (round (i / 2) rounding down)) rounding up
        make new symbol item in document 1 with properties ¬
            {symbol:symbol i of document 1, position:{100 + (the result * 150), (50 + i *
70)}} ¬

    end repeat
    set jobOptions to {class:job options, designation:all layers, reverse pages:true} ¬

    set colorOptions to {class:color management options, name:"ColorMatch RGB",
intent:saturation} ¬

    set coordinateOptions to {class:coordinate options, fit to page:true}
    set flatteningOptions to ¬
        {class:flattening options, clip complex regions:true, gradient resolution:60,
rasterization resolution:60} ¬

    set printOptions to ¬
        ¬
            {class:print options, job settings:jobOptions, color management
settings:colorOptions, coordinate settings:coordinateOptions, flattener
settings:flatteningOptions} ¬

    print document 1 options printOptions
end tell
```

quit

Forces Illustrator to quit.

Parameters	What it is	Objects supported	Returns
none		[application](#)	nothing

Quit Illustrator

```
-- Quit Illustrator after clearing the clipboard and closing documents
tell application "Adobe Illustrator"
    activate
    set the clipboard to {}
    close every document saving no
    quit
end tell
```

rasterize

Rasterizes the source art(s) within the specified clip bounds. The source art(s) is disposed of as a result of the rasterization.

Parameters	What it is	Objects supported	Returns
SourceArt **variant**	The page item(s) to be rasterized.	document	page item
ClipBounds **rect**	The rectangular region of the artwork for the rasterization. If this parameter is omitted, the bounds of the source art(s) is used instead.		
Options **rasterize options**	Describes the rasterization options.		

rearrange artboards

Rearranges artboards in a document.

Parameters	What it is	Objects supported	Returns
document	The `document` object or objects to operate on.	document	boolean
`[artboardLayout column/grid by column/grid by row/rl grid by row/rl row/row]`	The layout style to use. Default is `grid by row`.		
`[artboardRowsOrColumns integer]`	The number of rows or columns, as appropriate for the chosen layout style, in the range `[1..docNumArtboards-1]`, or 1 (the default) for single row/column layouts.		
`[artboardSpacing real]`	The spacing betweeen artboards. A number of pixels, default 20.0		
`[artboardMoveArtwork boolean]`	When `true` (the default), artwork is moved with the artboards.		

redo

Redoes the most recently undone transaction.

Parameters	What it is	Objects supported	Returns
none		[application](#)	nothing

redraw

Forces Illustrator to redraw its window or windows.

Parameters	What it is	Objects supported	Returns
none		application	nothing

Redraw

```
-- This script redraws all windows in Illustrator
tell application "Adobe Illustrator" to redraw
```

release tracing

Reverts vector artwork in the document that was created by tracing to the original source raster art, and removes the traced vector art. Returns the original object used to create the tracing, and deletes the `tracingobject` object and its associated `plugin item` object.

Parameters	What it is	Objects supported	Returns
tracingobject	The `tracingobject` object to operate on.	`tracingobject`	`placed item` or `raster item` object reference

relink

Relinks placed art in a document with the file that defines the content. Applied to a `placed item`, it associates the `placed item` object with the specified file.

Parameters	What it is	Objects supported	Returns
`file specification`	The file to be linked with the placed art.	`placed item`	nothing

Relink

```
-- This script relinks all placed art in the current document to a specific image file
tell application "Adobe Illustrator"
    activate

        set replacefile to (my (path to desktop folder) as string) & "replace.jpg"
        set placedItemList to every placed item of current document
        repeat with placedItemPath in placedItemList
                relink placedItemPath from replacefile
        end repeat
end tell
```

rotate

Rotates one or more page items counterclockwise by a specified rotation angle.

Parameters	What it is	Objects supported	Returns

`page item`	The `page item` object or objects to rotate.	`compound path item`
		`group item`
		`mesh item`
			nothing

`angle` **real**	The rotation angle in degrees. Rotation is counterclockwise.	`non native item`
		`page item`
		`path item`	

`[transforming objects` **boolean**`]`	If `true`, the page item positions and their orientations are affected. Default: `true`	`path point`
		`placed item`
		`plugin item`
		`raster item`	

`[transforming fill patterns` **boolean**`]`	If `true`, the fill patterns assigned to paths are affected. Default: `true`	`text frame`	
`[transforming fill gradients` **boolean**`]`	If `true`, the fill gradients assigned to paths are affected. Default: `true`		
`[transforming stroke patterns` **boolean**`]`	If `true`, the stroke patterns assigned to paths are affected. Default: `true`		

`[about`
  `document origin/`
  `top left/`
  `left/`
  `bottom left/`
  `top/`
  `center/`
  `bottom/`
  `top right/`
  `right/`
  `bottom right]`
	The point on the bounding box to which the rotation is applied. Default: `center`		

Notes   The `rotate` command provides many variations when used with the `about` parameter. Experiment with different choices for `about` to see what the results are for each setting.

Rotate about the bottom left corner

```
-- Rotate the first page item by 45 degrees using the
-- bottom left corner as the rotation pivot point
tell application "Adobe Illustrator"
    rotate page item 1 of document 1 angle 45.0 about bottom left
end tell
```

save

Saves an Illustrator document. Returns a reference to the saved document.

Parameters	What it is	Objects supported	Returns
`document`	The document to save.	document	object reference
[in `file specification`]	The file to save to, specified as a string containing the full file path or an alias. If not specified, the document is saved to its existing file.		

[as
  Illustrator/
  eps/
  pdf/
  fxg]
	The file type to which to save.		

[with options `anything`]	The save options for the specified file type.		

Save PDF files

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

```
-- Save each Illustrator file as a PDF file.
-- fileList is a list of aliases to Illustrator files
-- filePath is the path to the folder containing the files
-- destFolder is an alias to a folder where the PDF files are to be saved

on SaveFilesAsPDF(fileList, filePath, destFolder)
    set destPath to destFolder as string
    set fileCount to count of fileList
    if fileCount > 0 then
        repeat with i from 1 to fileCount
            set fileName to item i of fileList
            set fullPath to filePath & fileName
            set newFilePath to destPath & fileName & ".pdf"
            tell application "Adobe Illustrator"
                open POSIX file fullPath as alias without dialogs
                save current document in file newFilePath as pdf ¬
                    with options {class:PDF save options ¬
                    , compatibility:Acrobat 5 ¬
                    , preserve editability:true}
                close current document saving no
            end tell
        end repeat
    end if
end SaveFilesAsPDF
```

scale

Scales one or more page items by the specified horizontal and vertical amounts.

Parameters	What it is	Objects supported	Returns
page item	The `page item` object or objects to scale.	compound path item group item mesh item non native item page item path item path point placed item plugin item raster item text frame	nothing
horizontal scale **real**	The horizontal scaling factor. 100.0 is 100%		
vertical scale **real**	The vertical scaling factor. 100.0 is 100%		
[transforming objects **boolean**]	If `true`, the page item positions and their orientations are affected. Default: `true`		
[transforming fill patterns **boolean**]	If `true`, the fill patterns assigned to paths are affected. Default: `true`		
[transforming fill gradients **boolean**]	If `true`, the fill gradients assigned to paths are affected. Default: `true`		
[transforming stroke patterns **boolean**]	If `true`, the stroke patterns assigned to paths are affected. Default: `true`		
[line scale **real**]	The amount that line widths are to be scaled. 100.0 is 100%. Default: 100.0		

[about
  document origin/
  top left/
  left/
  bottom left/
  top/
  center/
  bottom/
  top right/
  right/
  bottom right]
	The point in the bounding box of the page item or items to which the scaling is applied. Default: `center`		

Notes The `scale` command provides many variations when used in conjunction with the `about` parameter. Experiment with different choices for the `about` parameter to see what the results are for each setting.

Scale a page item

```
-- Scale a page item by 50% horizontally resizing to the right
tell application "Adobe Illustrator"
    tell document 1
        scale page item 1 horizontal scale 50.0 vertical scale 100.0 about left
    end tell
end tell
```

select

Selects the text range.

Parameters	What it is	Objects supported	Returns
`text`	The `text` object or objects to select.	[text](#)	nothing
`[extending selection boolean]`	If `true`, the text range is added to the document's existing text selection. Default: `false`		

select objects on active artboard

Selects all art objects on the currently active artboard.

Parameters	What it is	Objects supported	Returns
`document`	The document object or objects to be operated upon.	`document`	`boolean`

select perspective preset

Selects a predefined perspective grid for the current document.

Parameters	What it is	Objects supported	Returns
document	The document object or objects to be operated upon.	document	boolean
perspective preset **text**	Name of a perspective grid preset.		

set

Changes the value of a variable or an object's property or data. This is a standard AppleScript command used to assign values to variables and object properties.

Parameters	What it is	Objects supported	Returns
property *or* **variable**	The object property or script variable to modify.	any property or variable	nothing
to **anything**	Any valid value.		

Set a property

```
-- Set the zoom property of the frontmost view window to 100%
tell application "Adobe Illustrator"
    set zoom of view 1 of document 1 to 1.0
end tell
```

set boolean preference

Sets the value of the application preference key as boolean. See "Preference accessor guidelines" on page 74.

Parameters	What it is	Objects supported	Returns
Illustrator preferences	The Illustrator preferences object or objects to be operated on.	Illustrator preferences	The boolean value of the preference key.
key **as Unicode text**	The type of data to retrieve.		

set integer preference

Sets the value of the application preference key as an integer. See <u>"Preference accessor guidelines" on page 74</u>.

Parameters	What it is	Objects supported	Returns
`Illustrator preferences`	The `Illustrator preferences` object or objects to be operated on.	<u>`Illustrator preferences`</u>	The integer value of the preference key.
`key` **`as Unicode text`**	The type of data to retrieve.		

set perspective active plane

Sets the active perspective plane for the active grid of the document.

Parameters	What it is	Objects supported	Returns
`document`	The document object or objects to be operated upon.	`document`	boolean

set real preference

Sets the value of the application preference key as a real number. See <u>"Preference accessor guidelines" on page 74</u>.

Parameters	What it is	Objects supported	Returns
`Illustrator preferences`	The `Illustrator preferences` object or objects to be operated on.	<u>Illustrator preferences</u>	The real value of the preference key.
`key` **`as Unicode text`**	The type of data to retrieve.		

set string preference

Sets the value of the application preference key as string type. See "Preference accessor guidelines" on page 74.

Parameters	What it is	Objects supported	Returns
Illustrator preferences	The Illustrator preferences object or objects to be operated on.	Illustrator preferences	The string value of the preference key.
key **as Unicode text**	The type of data to retrieve.		

show perspective grid

Shows the current active perspective grid for the document, if any; otherwise, shows the default perspective grid.

Parameters	What it is	Objects supported	Returns
`document`	The document object or objects to be operated upon.	document	boolean

show presets

Returns presets from a file as a list of Unicode text items.

Parameters	What it is	Objects supported	Returns
from **file specification**	The file to import from, specified as a string containing the full file path or an alias.	N/A	list (of Unicode text)

singular matrix

Tests an existing matrix to see if it is singular. A singular matrix cannot be inverted.

Parameters	What it is	Objects supported	Returns
`matrix`	The matrix to test.	`matrix`	`boolean`

Invert a matrix

```
-- This script gets an identity matrix and then
-- test to see if it can be inverted (if not singular)
-- If it can, then it inverts it
tell application "Adobe Illustrator"
    set someMatrix to get identity matrix
    if (not (singular matrix someMatrix)) then
        set someMatrix to invert matrix someMatrix
        set testResult to true
    else
        set testResult to false
    end if
end tell
```

store preset

Saves a set of preset tracing options from a `tracing options` object. For an existing preset, overwrites an unlocked preset and returns `true`. Returns `false` if the preset is locked.

Parameters	What it is	Objects supported	Returns
tracing options	The `tracing options` object to operate on.	tracing options	boolean
presetname **Unicode text**	The preset name. Use a name found in the `application` object's [tracing presets](#) list, or a new name to create a new preset.		

trace placed

Converts the raster art for the art item to vector art, using default options. Reorders the placed art into the source art of a plugin group, and converts it into a group of filled and/or stroked paths that resemble the original image.

Creates and returns a `plugin item` object that references a `traceobject` object.

Parameters	What it is	Objects supported	Returns
placed item object	The object to operate on.	`placed item`	`plugin item` object reference

trace raster

Converts the raster art for the art item to vector art, using default options. Reorders the raster art into the source art of a plugin group, and converts it into a group of filled and/or stroked paths that resemble the original image.

Creates and returns a `plugin item` object that references a `traceobject` object.

Parameters	What it is	Objects supported	Returns
`raster item object`	The object to operate on.	`raster item`	`plugin item` object reference

transform

Transform one or more page items by a specified matrix.

Parameters	What it is	Objects supported	Returns
page item	The `page item` object or objects to transform.	compound path item group item mesh item	nothing
using **matrix**	The matrix to use for the transformation.	non native item page item	
[transforming objects **boolean**]	If `true`, the page item positions and their orientations are affected. Default: `true`	path item path point placed item	
[transforming fill patterns **boolean**]	If `true`, the fill patterns assigned to paths are affected. Default: `true`	plugin item raster item text frame	
[transforming fill gradients **boolean**]	If `true`, the fill gradients assigned to paths are affected. Default: `true`		
[transforming stroke patterns **boolean**]	If `true`, the stroke patterns assigned to paths are affected. Default: `true`		
[line scale **real**]	The amount that line widths are to be scaled. Default: 100.0, which is 100%		

[about
  document origin/
  top left/
  left/
  bottom left/
  top/
  center/
  bottom/
  top right/
  right/
  bottom right]
	The point in the bounding box to which the transformation is applied. Default: `center`		

Notes This command can be used to generate any combination of transformations contained in a matrix, making it possible to skew objects among other modifications. The command provides many variations when used with the `about` parameter. Experiment with different choices for `about` to see what the results are for each setting.

Transform an object

```
-- This script skews an object 45 degrees to the right horizontally
-- by generating a rotation matrix and setting the appropriate matrix values
tell application "Adobe Illustrator"
    set baseMatrix to get rotation matrix angle 45.0
    set mvalue_b of baseMatrix to 0
    set startGeoBounds to geometric bounds of page item 1 of document 1
    transform page item 1 of document 1 using baseMatrix
end tell
```

translate

Moves one or more page items from their existing position in a document to a new position defined by relative coordinates.

Parameters	What it is	Objects supported	Returns
`page item`	The `page item` object or objects to translate.	`compound path item` `group item` `mesh item` `non native item` `page item` `path item` `path point` `placed item` `plugin item` `raster item` `text frame`	nothing
[delta x **real**]	The horizontal coordinate of the new position. Default: 0.0		
[delta y **real**]	The vertical coordinate of the new position. Default: 0.0		
[transforming objects **boolean**]	If `true`, the object positions and orientations are affected. Default: `true`		
[transforming fill patterns **boolean**]	If `true`, the fill patterns are affected. Default: `true`		
[transforming fill gradients **boolean**]	If `true`, the fill gradients are affected. Default: `true`		
[transforming stroke patterns **boolean**]	If `true`, the stroke patterns are affected. Default: `true`		

Notes Use `translate` to move objects relatively from their existing position. Set the `position` property of an object to move the object to absolute coordinates.

Move an item to a new position

```
--This script moves the first page item to new relative coordinates
tell application "Adobe Illustrator"
    set startGeoBounds to geometric bounds of page item 1 of document 1
    tell document 1 to translate page item 1 delta x 20.0 delta y -10.0
end tell
```

translate placeholder text

Translate the placeholder text to regular text. This allows you to enter Unicode characters as hex values.

Parameters	What it is	Objects supported	Returns
`Unicode text`	The placeholder text to be translated.	[text](#)	Unicode text or null

undo

Undoes the most recent transaction.

Parameters	What it is	Objects supported	Returns
none		<u>application</u>	nothing

update

Reapplies the dynamic data of the active dataset to the artboard.

Parameters	What it is	Objects supported	Returns
dataset	Dataset to be updated.	dataset	dataset