PowerPoint 2004 AppleScript Reference

Using the PowerPoint Dictionary

Microsoft Office Suite Classes

Microsoft Office Suite Commands

Microsoft PowerPoint Suite Classes

Microsoft PowerPoint Suite Commands

Drawing Suite Classes

Drawing Suite Commands

Text Suite Classes

Text Suite Commands

Table Suite Classes

Table Suite Commands

Using the PowerPoint Dictionary

About the PowerPoint AppleScript Dictionary

The Microsoft PowerPoint AppleScript Dictionary provides comprehensive access to the classes and commands you need for programmatically controlling PowerPoint by using AppleScript. The PowerPoint AppleScript Dictionary is based on the Visual Basic object model for PowerPoint, and as such its terminology and syntax closely mirror the terminology and syntax used in Visual Basic. As a result of this close relationship with Visual Basic, the PowerPoint AppleScript Dictionary might appear more complex and less "English-like" than other AppleScript Dictionaries.

Using classes and commands

Classes and commands in the PowerPoint AppleScript Dictionary often have many properties and parameters. Many of these properties and parameters are represented by enumerations — or lists — of constants rather than text or numbers or Boolean operators. These enumerations use many of the same words as their parameters or properties and are usually descriptive of what they represent. Because of the nature of the AppleScript language and the complexity of the underlying object model, expressions containing several properties and parameters can be very long and often look confusing. In such cases, it is helpful to use parentheses to separate out one property from another to make your scripts easier to read. Additionally, you will find that the PowerPoint object model is very deep, requiring several layers to access certain objects. You might find it helpful to use variables to store these objects and reduce the complexity of your script.

As in other applications, optional command parameters are denoted in the dictionary by square brackets [ ]. Class properties always have default values, which means that when you create new objects it is not necessary to supply values for all of the properties. Properties denoted by [r/o] are read-only and cannot be set once an object exists. However, when you use the **make** command to create new objects, most read-only properties can be set by using the **with properties** clause. For more information about the properties available for various objects, see the object's entry in the Dictionary Reference portion of this guide.

Working with shapes (drawing objects)

Shapes, or drawing objects, are represented by the **shape** object. Shapes on slides are numbered sequentially, regardless of the shape type. For example, if a slide has a text box, a place holder and an AutoShape, you can refer to those shapes as follows:

```
text box 1 of slide 1 of active presentation
```

```
place holder 2 of slide 1 of active presentation
```

```
shape 3 of slide 1 of active presentation
```

Setting properties for a shape

Many formatting properties of shapes aren't set by properties that apply directly to the **shape** object. Instead, related shape attributes are grouped under secondary objects, such as the **fill format** object, which contains all the properties that relate to the shape's fill, or the **line format** object, which contains all the properties that are unique to the border around the object. To set properties for a shape, you must first return the object that represents the set of related shape attributes and then set properties of that returned object. For example, you use the *fill* property to return the **fill format** object, and then you set the *fore color* property of the **fill format** object to set the fill foreground color for the specified shape, as shown in the following example.

```
set myDocument to slide 1 of active presentation
set fore color of fill of shape 1 of myDocument to ({255, 0, 0} as RGB color)
```

Applying a property or command to several shapes at the same time

In the user interface, you can perform some operations with several shapes selected; for example, you can select several shapes and set all their individual fills at once. There are other operations you can perform only with a single shape selected; for example, you can edit only the text in a shape if a single shape is selected.

If you can perform an operation on multiple selected shapes in the user interface at the same time, you can often do the programmatic equivalent by applying the appropriate properties and commands to the **shapes** list. The following example applies a gradient fill to all of the shapes on slide 1 of the presentation.

```
set myDocument to slide 1 of active presentation
set myShapes to (get shapes of myDocument)
preset gradient myShapes style horizontal gradient variant 1 ¬
        gradient type gradient brass
```

Even if you cannot perform an operation on several shapes in the user interface at the same time by selecting them and then using a command, you can perform the equivalent action programmatically by looping through the **shapes** list that contains the shapes you want to work with, and applying the appropriate properties and commands to the individual **shape** objects in the list. The following example loops through all the shapes on myDocument and adds text to each shape that is an AutoShape.

```
set myDocument to slide 1 of active presentation
repeat with sh in (get shapes of myDocument)
    if shape type of sh is shape type auto then
        insert the text " (version 1)" at the end of ¬
            text range of text frame of sh
    end if
end repeat
```

8

Microsoft Office Suite

Microsoft Office Suite Classes

Class: assistant

Plural

assistants

Elements

balloon

Represents the Microsoft Office Assistant.

Use the *office assistant* property of the application class to return the **assistant**. Only one **assistant** object can be active at a time. Use the *visible* property to display the Assistant.

By default, the Office Assistant is off. The default Assistant is Max. To select a different Assistant programatically, use the *file name* property.

The following example turns on, displays, and animates the Assistant.

```
set assistant on of office assistant to true

set visible of office assistant to true

set animation of office assistant to greeting
```

Properties

animation

Returns or sets an animation action for the Office Assistant. When this property is applied to the **assistant** object, the Assistant is animated immediately (if it is visible). When this property is applied to the **balloon** object, the Assistant is animated only while the balloon is displayed.

Can be one of the following constants:

appear	**idle**
begin speaking	**listens to computer**
character success major	**look down**
checking something	**look down left**
disappear	**look down right**
empty trash	**look left**
gesture down	**look right**
gesture left	**look up**
gesture right	**look up left**
gesture up	**look up right**
get artsy	**printing**
get attention major	**saving**
get attention minor	**searching**
get techy	**sending mail**
get wizardy	**thinking**
goodbye	**working at something**
greeting	**writing noting something**

assist with help

True if the Office Assistant appears when the user presses the HELP key to display Help. Read/write.

If this property is set to **False**, the Help window appears instead of the Office Assistant.

This property corresponds to the **Respond to HELP key** option (**Office Assistant** dialog box, **Options** tab).

assist with wizards

True if the Office Assistant provides online Help with wizards. Read/write.

This property corresponds to the **Help with wizards** option (**Office Assistant** dialog box, **Options** tab).

assistant item

Returns the text associated with the Office Assistant. Read-only.

assistant on

True if the Office Assistant is enabled. Read/write.

balloon error

Returns a value that indicates the last recorded balloon error. Read-only.

Can be one of the following:

bad picture reference	The balloon contains a graphic that couldn't be displayed because the file doesn't exist or because the graphic isn't a valid PICT file.
bad reference	The balloon contains an unrecognized or unsupported reference.
buttonless modal	The balloon you attempted to display is modal, but it contains no buttons. The balloon won't be shown because it can't be dismissed.
button modeless	The balloon you attempted to display is modeless, contains no buttons, and has no procedure assigned to the *callback* property. The balloon won't be shown because a callback procedure is required for modeless balloons.
none	No error was encountered.
bad character	The balloon contains an ASCII control character other than CR or LF and greater than 32.
out of memory	The balloon won't appear because there is insufficient memory.
too big	The balloon is too big to appear on the screen.
other	The balloon won't appear because some other error occurred; for example, another modal balloon is already active.

feature tips

True if the Office Assistant provides information about using application features more effectively. Read/write.

This property corresponds to the **Using features more effectively** option (**Office Assistant** dialog box, **Options** tab).

file name

Returns or sets the name of the file for the active Office Assistant. Read/write.

The Office Assistant files are installed in the Microsoft Office 2004/Office/Assistants folder.

keyboard shortcut tips

True if the Office Assistant displays Help about keyboard shortcuts. Read/write.

This property corresponds to the **Keyboard shortcuts** option (**Office Assistant** dialog box, **Options** tab).

left position

Returns the horizontal position of the Office Assistant window (in points) relative to the screen.

mouse tips

> **True** if the Office Assistant provides suggestions for using the mouse effectively. Read/write.
>
> This property corresponds to the **Using the mouse more effectively** option (**Office Assistant** dialog box, **Options** tab).

move when in the way

> **True** if the Office Assistant window automatically moves when it's in the way of the user's work area. For example, the Office Assistant will move if it's in the way of dragging or dropping or in the way of keystroke entries. Read/write.
>
> This property corresponds to the **Move when in the way** option (**Office Assistant** dialog box, **Options** tab).

name

> Returns the name of the Office Assistant currently in use. Read-only.

sounds

> **True** if the Office Assistant produces the sounds that correspond to animations. Read/write.
>
> If a sound card is not installed, this property has no effect.

tip of the day

> **True** if the Office Assistant displays a special tip each time the Office application is opened. Read/write.
>
> This property corresponds to the **Show the Tip of the Day at startup** option (**Office Assistant** dialog box, **Options** tab).

top

> Returns the vertical position of the Office Assistant window (in points) relative to the screen..

visible

> **True** if the Office Assistant is visible. Read-write.

Class: balloon checkbox

Represents a check box in the Office Assistant balloon.

Use **balloon checkbox** *index*, where *index* is a number from 1 through 5, to return a single **balloon checkbox** object. There can be up to five check boxes in one balloon; each check box appears when a value is assigned to its *checkbox text* property.

Remarks

Balloon check boxes display the user's choices until the user dismisses the balloon. To record the user's choice as soon as the user clicks the button beside the label, see the topic on balloon labels.

Properties

balloon checkbox item

> Returns the text associated with the balloon check box. Read-only.

checkbox text

> Returns or sets the text displayed next to a check box in the Office Assistant balloon. Read/write.

checked

> **True** if the specified check box in the Office Assistant balloon is checked. Read/write.

name

> Returns the name of the check box label. Read-only.

Class: balloon

Plural

balloons

Elements

balloon checkbox

balloon label

Represents the balloon in which the Office Assistant displays information. A balloon can contain controls such as check boxes and labels.

Only one balloon can be visible at a time. However, it's possible to define several balloons and display any one of them when needed. For more information, see "Defining and Reusing Balloons" later in this topic.

To make the specified balloon visible, use the show balloon command. Use the *call back* property to run procedures based on selections from modeless balloons (balloons that remain visible while a user works in the application).

The following example creates a balloon that contains tips for saving entered data.

```
set newbal to make new balloon of office assistant
set balloon type of newbal to bullets
set icon of newbal to icon tip
set balloon button of newbal to buttons ok cancel
set heading of newbal to "Tips for Saving Information."
repeat 3 times
    make new balloon label at the beginning of newbal
end repeat
set label text of balloon label 1 of newbal to "Save your work often."
set label text of balloon label 2 of newbal to "Install a surge protector."
set label text of balloon label 3 of newbal to "Exit your application properly."
show balloon newbal
```

Defining and Reusing Balloons

You can reuse balloon objects you've already created by assigning the object to a variable and displaying the variable when you need it. This example defines `balloon1`, `balloon2`, and `balloon3` separately so they can be reused.

```
set balloon1 to make new balloon of office assistant
set heading of balloon1 to "First balloon"


Set balloon2 to make new balloon of office assistant
set heading of balloon2 to "Second balloon"


Set balloon3 to make new balloon of office assistant
set heading of balloon3 to "Third balloon"


show balloon balloon1
show balloon balloon2
show balloon balloon3
```

Alternatively, instead of using separate variables, you can place the balloon object into an array.

Properties

balloon button

> Returns or sets the type of button displayed at the bottom of the Office Assistant balloon. When you create a new balloon, this property is initially set to **OK**. Read/write.
>
> Can be one of the following:

buttons abort retry ignore	**buttons ok cancel**
buttons back close	**buttons retry cancel**
buttons back next close	**buttons search close**
buttons back next snooze	**buttons tips options close**
button cancel	**buttons yes all no cancel**
buttons next close	**buttons yes no cancel**
button none	**buttons yes no**
button ok	

balloon mode

Returns or sets the type of balloon displayed. Read/write.

Can be one of the following:

- **auto down balloon** - Balloon is instantly dismissed when the user clicks anywhere on the screen.

- **modal balloon** - This is the initial default setting. User must dismiss the balloon before he or she can return to working in the application.

- **modeless balloon** - User can work in the application while the balloon is visible. When this value is used, a value for the *call back* callback property is also required. When you create a new balloon, this property is initially set to **modal balloon**.

balloon text

Returns or sets the text displayed after the heading but before the labels or check boxes in the Office Assistant balloon. Read/write.

You can specify a graphic to display by using the following syntax:

{type location sizing_factor},
where
type is pict (Macintosh PICT file).
location is the resource id or the path and file name.
sizing_factor specifies the width of the pict.

balloon type

Returns or sets the type of balloon the Office Assistant uses. Read/write.

Can be one of the following:

- **buttons** - This is the initial default setting.

- **bullets**

- **numbers**

call back

Sets the name of the procedure to run from a modeless balloon. Read/write String.

The procedure you specify for the *call back* property must be written to receive three arguments:

- The balloon object that called the procedure

- The value of the button the user clicked

- An integer that uniquely identifies the balloon that called the procedure

heading

> Returns or sets the heading that appears in the Office Assistant balloon. Read/write.

> You can specify a graphic to display in the balloon heading by using the following syntax:

> **{***type location sizing_factor***}**
> where
> *type* is pict (Macintosh PICT file).
> *location* is the resource id or the path and file name
> *sizing_factor* specifies the width of the pict.

icon

> Returns or sets the type of icon that appears in the upper-left portion of the Office Assistant balloon. Read/write.

> Can be one of the following:

> - **icon alert**
> - **icon alert critical**
> - **icon alert info**
> - **icon alert warning**
> - **icon application**
> - **icon none**
> - **icon tip**

name

> Returns the name of the balloon. Read-only.

Class: balloon label

Represents a label in the Office Assistant balloon.

Use **balloon label** *index*, where *index* is a number from 1 through 5, to return a **balloon label** object. There can be up to five labels on one balloon; each label appears when a value is assigned to its *label text* property.

Remarks

Balloon labels record the user's choice as soon as the user clicks the button beside the label. To display the user's choices until the user dismisses the balloon, see the topic on balloon check boxes.

Properties

balloon label item

> Returns the text associated with the balloon label. Read-only.

label text

> Returns or sets the text displayed next to a label in the Office Assistant balloon. Read/write.

> **Remarks**

> You can specify a graphic to display by using the following syntax:

> {type location sizing_factor},
> where
> *type* is pict (Macintosh PICT file)
> *location* is the resource id or the path and file name
> and *sizing_factor* specifies the width of the pict

name

> Returns the name of the balloon label. Read-only.

Class: command bar

Plural

command bars

Elements

command bar control

Represents a command bar in an Office application.

Use **command bar** *index*, where *index* is the name or index number of a command bar, to return a **command bar** object. The following example steps through the collection of command bars to find the command bar named "Forms." If it finds this command bar, the example makes it visible and protects its docking state. In this example, the variable *bars* represents the list of all **command bars**, and the variable *cb* represents a **command bar** object.

```
set foundFlag to false
set bars to command bars
repeat with cb in bars
     repeat 1 times
          if name of cb is "Forms" then
               set protection of cb to no change dock
               set visible of cb to true
               set foundFlag to true
          end if
          exit repeat
          set end of bars to cb
     end repeat
end repeat
if foundFlag is false then
     display dialog "There is no Forms command bar."
end if
```

You can use a name or index number to specify a menu bar or toolbar in the list of available menu bars and toolbars in the Office application. However, you must use a name to specify a menu, shortcut menu, or submenu (all of which are represented by **command bar** objects). This example adds a new menu item to the bottom of the **Tools** menu. When a user clicks the new menu item, it runs the procedure named "qtrReport."

```
set newItem to make new command bar button at command bar "Tools"

set begin group of newItem to True

set caption of newItem to "Make Report"

set face id of newItem to 0

set OnAction of newItem to "qtrReport"
```

If two or more custom menus or submenus have the same name, **command bar** *index* returns the first one.

Properties

bar position

> Returns or sets the position of a command bar. Read/write.
>
> Can be one of the following:

- **bar left**
- **bar top**
- **bar right**
- **bar bottom**
- **bar floating**
- **bar pop up**
- **bar menu**

bar type

> Returns the type of command bar. Read-only.
>
> Can be one of the following:

- **normal command bar**
- **menubar command bar**
- **popup command bar**

built in

> **True** if the specified command bar or command bar control is a built-in command bar or control of the container application. **False** if it's a custom command bar or control. Read-only.

context

Returns or sets a string that determines where a command bar will be saved. The string is defined and interpreted by the application. Read/write.

You can set the *context* property only for custom command bars. This property will fail if the application doesn't recognize the context string, or if the application doesn't support changing context strings programmatically.

enabled

True if the specified command bar or command bar control is enabled. Read/write.

For command bars, setting this property to **True** causes the name of the command bar to appear in the list of available command bars.

For built-in controls, setting this property to **True** causes the application to determine the state of the control. Setting this property to **False** forces the control to be disabled.

entry index

Returns the index number for a command bar. Read-only.

The position of the first command bar control is 1. Separators are not counted.

height

Returns or sets the height (in pixels) of a command bar or command bar control. Read/write.

An error will occur if you attempt to set the *height* property for a command bar that isn't in a resizable state (that is, if it's docked or protected from resizing).

left position

Returns or sets the distance (in pixels) from the left edge of the specified command bar or command bar control to the left edge of the screen. Returns the distance from the left side of the docking area. Read/write.

local name

Returns the name of a built-in command bar as it's displayed in the language version of the container application, or returns or sets the name of a custom command bar. Read/write.

Note   If you attempt to set this property for a built-in command bar, an error occurs.

The local name of a built-in command bar is displayed in the title bar (when the command bar isn't docked) and in the list of available command bars, wherever that list is displayed in the container application.

If you change the value of the *local name* property for a custom command bar, the value of *name* changes as well, and vice versa.

name

Returns or sets the name of the specified object. Read/write.

The local name of a built-in command bar is displayed in the title bar (when the command bar isn't docked) and in the list of available command bars, wherever that list is displayed in the container application.

For a built-in command bar, the *name* property returns the command bar's U.S. English name. To return the localized name, use the *local name* property.

If you change the value of the *name* property for a custom command bar, the value of *local name* changes as well, and vice versa.

protection

Returns or sets the way a command bar is protected from user customization. Read/write.

Can be one of or a sum of the following:

- **no protection**
- **no customize**
- **no resize**
- **no move**
- **no change visible**
- **no change dock**
- **no vertical dock**
- **no horizontal dock**

row index

Returns or sets the docking order of a command bar in relation to other command bars in the same docking area. Can be an integer greater than zero.

Several command bars can share the same row index, and command bars with lower numbers are docked first. If two or more command bars share the same row index, the command bar most recently assigned will be displayed first in its group.

top

Returns or sets the distance (in pixels) from the top edge of the specified command bar or command bar control to the top edge of the screen. For docked command bars, this property returns or sets the distance from the command bar to the top of the docking area. Read/write.

visible

True if a command bar or command bar control is visible. Read/write.

width

Returns or sets the width (in pixels) of the specified command bar or command bar control. Read/write.

Class: command bar button

Plural

command bar buttons

Represents a button control on a command bar.

Use **command bar control** *index*, where *index* is the index number of the control, to return a **command bar button** object. (The *control type* property of the control must be **control button**.)

Properties

button face is default

> **True** if the face of the specified command bar button control is its original built-in face. This property can only be set to **True**, which will reset the face to the built-in face. Read/write.

button state

> Returns or sets the appearance of a command bar button control. Can be one of the following

- **button state up**
- **button state down**
- **button state unset**

button style

> Returns or sets the way a command bar button control is displayed. Read/write.
>
> Can be one of the following:

- **button automatic**
- **button icon**
- **button caption**
- **button icon and caption**

face id

> Returns or sets the ID number for the face of a command bar button control. Read/write.
>
> The *face id* property dictates the look, but not the function, of a command bar button. To determine the function of the button, use the *id* property of the **command bar control** object.
>
> The value of the *face id* property for a command bar button with a custom face is 0 (zero).

shortcut text

> Returns or sets the shortcut key text displayed next to a button control when the button appears on a menu, submenu, or shortcut menu. Read/write.
>
> You can set this property only for command bar buttons that contain an on-action macro.

Class: command bar combobox

Plural

command bar comboboxes

Represents a combo box control on a command bar.

Use **command bar control** *index*, where *index* is the index number of the control, to return a **command bar combobox** object. (The *control type* property of the control must be **control edit**, **control dropdown**, **control combobox**, **button dropdown**, **split dropdown**, **OCX dropdown**, **graphic combo**, or **graphic dropdown**.)

The following example creates a new command bar combobox and adjusts the size of the control on the command bar named "Custom," and then it adds two items to the combo box.

```
set newCombo to make new command bar control at command bar "Custom" with ¬
        properties {control type:control combobox}
set drop down lines of newCombo to 3
set drop down width of newCombo to 90
set list index of newCombo to 0
add item to combobox newCombo combobox item "First Item" entry index 1
add item to combobox newCombo combobox item "Second Item" entry index 2
```

Properties

combobox style

> Returns or sets the way a command bar combo box control is displayed. Read/write.
>
> Can be either of the following:
>
> - **combobox style label**
>
> - **combobox style normal**

combobox text

> Returns or sets the text in the display or edit portion of the command bar combo box control. Read/write.

drop down lines

> Returns or sets the number of lines in the specified command bar combo box control. The combo box control must be a custom control, and it must be either a drop-down list box or a combo box. Read/write.
>
> **Note**   If this property is set for a combo box control that's either an edit box or a built-in combo box control, an error occurs.
>
> If this property is set to 0 (zero), the number of lines in the control will be based on the number of items in the list.

drop down width

> Returns or sets the width (in pixels) of the list for the specified command bar combo box control. Read/write.

> **Note**   An error occurs if you attempt to set this property for a built-in control.

> If this property is set to -1, the width of the list is based on the length of the longest item in the combo box list. If this property is set to 0, the width of the list is based on the width of the control.

list index

> Returns or sets the index number of the selected item in the list portion of the command bar combo box control. If nothing is selected in the list, this property returns zero. Read/write.

> **Note**   This property fails when applied to controls other than list controls.

> Setting the list index property causes the specified control to select the given item and execute the appropriate action in the application.

Class: command bar control

Plural

command bar controls

Represents a command bar control. The properties and commands of the **command bar control** object are all shared by the **command bar button**, **command bar combobox**, and **command bar popup** classes.

Note   To write scripts to work with custom command bar controls, use the **command bar button**, **command bar combobox**, and **command bar popup** classes. To write scripts to work with built-in controls in the container application that cannot be represented by one of those three classes, use the **command bar control** class.

Use **command bar control** *index*, where *index* is the index number of a control, to return a **command bar control** object. (The *control type* property of the control must be **control label**, **expanding grid**, **split expanding grid**, **control grid**, or **control gauge**.)

Note   Variables declared as **command bar control** can be assigned **command bar button**, **command bar combobox**, and **command bar popup** values.

Properties

begin group

> **True** if the specified command bar control appears at the beginning of a group of controls on the command bar. Read/write.

built in

> **True** if the specified command bar or command bar control is a built-in command bar or control of the container application. **False** if it's a custom command bar or control, or if it's a built-in control whose *on action* property has been set. Read-only.

control type

Returns the type of command bar control. Read-only.

Can be one of the following:

control button	**graphic popup**
button drop down	**graphic dropdown**
button popup	**control grid**
control combobox	**control label**
control custom	**OCX dropdown**
control dropdown	**msopopup**
control edit	**split button MRU popup**
expanding grid	**split button popup**
control gauge	**split dropdown**
generic dropdown	**split expanding grid**
graphic combo	

description text

Returns or sets the description for the specified command bar control. The description is displayed in the status bar of the container application when the user positions the pointer over a command bar control. Read/write.

Not all applications display a status bar.

enabled

True if the specified command bar or command bar control is enabled. Read/write.

For command bars, setting this property to **True** causes the name of the command bar to appear in the list of available command bars.

For built-in controls, if you set the *enabled* property to **True**, the application determines its state. Setting the *enabled* property to **False** forces the control to be disabled.

entry index

Returns the index number for the **command bar control**. Read-only.

height

Returns or sets the height (in pixels) of a command bar or command bar control. Read/write.

An error will occur if you attempt to set the height property for a command bar that isn't in a resizable state (that is, if it's docked or protected from resizing).

help context ID

Returns or sets the Help context Id number for the Help topic attached to the command bar control. Read/write.

To use this property, you must also set the *help file* property.

help file

Returns or sets the Help file name for the Help topic attached to the command bar control. Read/write.

To use this property, you must also set the *help context ID* property.

id

Returns the ID for a built-in command bar control. Read-only.

A control's ID determines the built-in action for that control. The value of the *id* property for all custom controls is 1.

left position

Returns the distance (in pixels) from the left edge of the specified command bar or command bar control to the left edge of the screen. Returns the distance from the left side of the docking area. Read-only.

name

Returns or sets the caption text for a command bar control. Read/write.

on action

Returns or sets the name of a Visual Basic procedure that will run when the user clicks or changes the value of a command bar control. Read/write.

parameter

Returns or sets a string that an application can use to execute a command. Read/write.

If the specified parameter is set for a built-in control, the application can modify its default behavior if it can parse and use the new value. If the parameter is set for custom controls, it can be used to send information to Visual Basic procedures, or it can be used to hold information about the control (similar to a second *tag* property value).

priority

Returns or sets the priority of a command bar control. A control's priority determines whether the control can be dropped from a docked command bar if the command bar controls can't fit in a single row. Read/write.

Valid priority numbers are 0 (zero) through 7. Special priority numbers are 0 and 1. A priority of 0 indicates an "automatic" value, which means a number is assigned based on the type of control. A priority of 1 means the control cannot be dropped.

tag

Returns or sets information about the command bar control, such as data that can be used as an argument in procedures, or information that identifies the control. Read/write.

tooltip text

Returns or sets the text displayed in a command bar control's ScreenTip. Read/write.

top

> Returns or sets the distance (in pixels) from the top edge of the specified command bar or command bar control to the top edge of the screen. For docked command bars, this property returns or sets the distance from the command bar to the top of the docking area. Read-only.

visible

> **True** if a command bar or command bar control is visible. Read/write.

width

> Returns or sets the width (in pixels) of the specified command bar or command bar control. Read/write.

Class: command bar popup

Plural

command bar popups

Elements

command bar control

Represents a pop-up control on a command bar.

Use **command bar control** *index*, where *index* is the number of the control, to return a **command bar popup** object. (The **control type** property of the control must be **control popup**, **graphic popup**, **button popup**, **split button popup**, or **split button MRU popup**.)

Class: custom document property

Plural

custom document properties

Represents a custom document property of a container document.

Use **custom document property** *index*, where *index* is the name or index number of the custom document property, to return a **custom document property** object that represents a specific custom document property.

Class: document property

Plural

document properties

Represents a built-in document property of a container document.

Use **document property** *index*, where *index* is the name or index number of the built-in document property, to return a single **document property** object that represents a specific built-in document property.

The names of all the available built-in document properties are shown on the following list:

Title	**Number of Words**
Subject	**Number of Characters**
Author	**Security**
Keywords	**Category**
Comments	**Format**
Template	**Manager**
Last Author	**Company**
Revision Number	**Number of Bytes**
Application Name	**Number of Lines**
Last Print Date	**Number of Paragraphs**
Creation Date	**Number of Slides**
Last Save Time	**Number of Notes**
Total Editing Time	**Number of Hidden Slides**
Number of Pages	**Number of Multimedia Clips**

Container applications don't necessarily define a value for every built-in document property. If a given application doesn't define a value for one of the built-in document properties, returning the *value* property for that document property causes an error.

Properties

document property type

> Returns or sets the document property type. Read-only for built-in document properties; read/write for custom document properties.

> Can be one of the following:

> - **boolean**

> - **date**

> - **float**

> - **number**

> - **string**

link source

> Returns or sets the source of a linked custom document property. Read/write.

> This property applies only to custom document properties; you cannot use it with built-in document properties.

> The source of the specified link is defined by the container application.

> Setting the *link source* property sets the *link to context* property to **True**.

link to context

> **True** if the value of the custom document property is linked to the content of the container document. **False** if the value is static. Read/write.

> This property applies only to custom document properties. For built-in document properties, the value of this property is **False**.

> Use the *link source* property to set the source for the specified linked property. Setting the *link source* property sets the *link to context* property to **True**.

name

> Returns the name of the specified object. Read-only.

value

> Returns or sets the value of a document property. Read/write.

> If the container application doesn't define a value for one of the built-in document properties, reading the *value* property for that document property causes an error.

Class: web page font

Represents the default font used when documents are saved as Web pages for a particular character set.

Use the **web page font** object to describe the proportional font, proportional font size, fixed-width font, and fixed-width font size for any available character set. The following character sets are supported:

Arabic	Korean
Cyrillic	Multilingual
English Western European Other Latin	Unicode
Script	Simplified Chinese
Greek	Thai
Hebrew	Traditional Chinese
Japanese	Vietnamese

Properties

fixed width font

> Sets or returns the fixed-width font setting in the host application. Read/write.

> When you set the *fixed width font* property, the host application does not check the value for validity.

fixed width font size

> Sets or returns the fixed-width font size setting in the host application, in points. Read/write.

> When you set the *fixed width font size* property, the host application does not check the value for validity. If you enter an invalid value, such as a nonnumber, the host application sets the size to 0 points. You can enter half-point sizes; if you enter other fractional point sizes, they are rounded up or down to the nearest half-point.

proportional font

> Sets or returns the proportional font setting in the host application. Read/write.

> When you set the *proportional font* property, the host application does not check the value for validity.

proportional font size

> Sets or returns the proportional font size setting (in points) in the host application. Read/write.

> When you set the *proportional font size* property, the host application does not check the value for validity. If you enter an invalid value, such as a nonnumber, the host application sets the size to 0 points. You can enter half-point sizes; if you enter other fractional point sizes, they are rounded up or down to the nearest half-point..

Microsoft Office Suite Commands

Command: add item to combobox

Adds a list item to the specified command bar combo box control. The combo box control must be a custom control, and it must be either a drop-down list box or a combo box.

Note This command will fail if it's applied to an edit box or a built-in combo box control.

Syntax

add item to combobox command bar combobox Required. An expression that returns a command bar combobox object

  combobox item Unicode text Required. The text to be added to the specified control.

  [**entry index** small integer] Optional. The position of the specified item in the list. If this argument is omitted, the item is added at the end of the list.

Example

This example adds a combo box control to a command bar. Two items are added to the control, and the number of line items and the width of the combo box are set.

```
set myBar to make new command bar with properties ¬

        {bar type:normal command bar, name:"New"}

set myControl to make new command bar control at myBar¬

    with properties {control type:control combobox, combobox text:"Monday",¬

    combobox style:combobox style label, drop down lines:2, drop down width:75}

add item to combobox myControl combobox item "Monday" entry index 1

add item to combobox myControl combobox item "Tuesday" entry index 2
```

Command: `clear combobox`

Removes all list items from the specified command bar combo box control (drop-down list box or combo box) and clears the text box (edit box or combo box).

Note   This command will fail if it's applied to a built-in command bar control.

Syntax

clear combobox command bar combobox   Required. An expression that returns a **command bar combobox** object.

Example

This example checks the number of items in the combo box control named "Names" on the command bar named "Custom." If there are more than three items in the list, the example clears the list, adds a new first item to the list, and displays this new item as the default for the combo box control.

```
set myBar to command bar "Custom"

set myControl to command bar control "Names" of myBar

set listItems to get count of combobox items myControl

if listitems > 3 then

    clear combobox myControl

    add item to combobox myControl combobox item "Third Item" entry index 1

    set list index of myControl to 1

end if
```

Command: `execute`

Runs the procedure associated with a command bar control.

Syntax

execute   command bar control   Required. An expression that returns a **command bar control**, **command bar button**, or **command bar combobox** object.

Remarks

This command runs the specified script or command if controls are currently enabled in the application and if the *enabled* property of the specified object is **True**.

Applying this command to a **command bar popup** object generates a run-time error.

Example

This example checks the value of the combo box control on the custom command bar named "My Custom Bar." If the index number of the command bar control is 1, the example runs the script specified by the *on action* property of the command bar control.

```
set mycontrol to the first command bar combobox of command bar "My Custom Bar"

if entry index of myControl = 1 then

    execute myControl

end if
```

Command: `get combobox item`

Returns the string at the given index of items within a combo box control.

Syntax

get combobox item   command bar combobox   Required. An expression that returns a **command bar combobox** object.

   entry index   small integer   Required. The position of the specified item in the list. If this argument is omitted, the last item in the list is returned.

Command: `get count of combobox items`

Returns the number of items within a combo box control.

Syntax

get count of combobox items   command bar combobox   Required. An expression that returns a **command bar combobox** object.

Example

This example uses the **get count of combobox items** command to check the number of items in the combo box control named "Names" on the command bar named "Custom." If there are more than three items on the list, the example clears the list, adds a new first item to the list, and displays this new item as the default for the combo box control.

```
set myBar to command bar "Custom"

set myControl to command bar control "Names" of myBar

set listItems to get count of combobox items myControl

if listitems > 3 then

    clear combobox myControl

    add item to combobox myControl combobox item "Third Item" entry index 1

    set list index of myControl to 1

end if
```

Command: remove an item from combobox

Removes an item from a command bar combo box control.

Note   This command fails when applied to controls other than list controls.

Syntax

remove an item from combobox   command bar combobox   Required. An expression that returns a **command bar combobox** object.

   entry index   small integer   Required. The index number of the item to be removed from the list.

Example

The following example determines whether there are more than three items in a combo box control. If there are more than three items, the example removes the second item, alters the style, and sets a new value.

```
set myBar to command bar "Custom"
set myControl to command bar combobox 1 of myBar
set listitems to get count of combobox items myControl
if listitems > 3 then
    remove an item from combobox myControl entry index 2
    set combobox style of myControl to combobox style normal
    set combobox text of myControl to "New Default"
end if
```

Command: reset

Resets a built-in command bar to its default configuration, or resets a built-in command bar control to its original function and face.

Syntax

reset   command bar/command bar control   Required. An expression that returns a **command bar**, **command bar control**, **command bar button**, **command bar popup**, or **command bar combobox** object.

Remarks

Resetting a built-in control restores the actions originally intended for the control and resets each of the control's properties back to its original state. Resetting a built-in command bar removes custom controls and restores built-in controls.

Example

This example resets the **Formatting** toolbar to its default state.

```
reset command bar "Formatting"
```

Command: `show balloon`

Displays the specified balloon object. Returns a constant that indicates which balloon the user clicks.

Syntax

show balloon balloon Required. An expression that returns a **balloon** object.

Remarks

You can use the return value of the **show balloon** command to display a user's button selection. The **show balloon** command returns one of the following buttons:

abort button	**back button**
cancel button	**close button**
ignore button	**next button**
no button	**null button**
OK button	**options button**
retry button	**search button**
snooze button	**tips button**
yes button	**yes to all button**

Example

This example creates a balloon that contains three choices.

```
set b to make new balloon at office assistant
tell b
    set heading to "This is my heading"
    set balloon text to "Select one of these things"
    repeat 3 times
        make new balloon label at the beginning
    end repeat
    set label text of balloon label 1 to "Choice One"
    set label text of balloon label 2 to "Choice Two"
    set label text of balloon label 3 to "Choice Three"
    show balloon
end tell
```

Microsoft PowerPoint Suite

Microsoft PowerPoint Suite Classes

Class: action setting

Contains information about how the specified shape or text range reacts to mouse actions during a slide show. The **action setting** object represents how the specified object reacts when the user clicks it during a slide show or the user moves the mouse pointer over it during a slide show.

Use the **get action setting for** command to return the **action setting** object for a shape or the **get text action setting** command to return the **action setting** object for a text range. The following example sets the mouse-click action for the text in the third shape on slide one in the active presentation to an Internet link.

```
set theAction to (get text action setting text range of text frame of ¬
        shape 3 of slide 1 of active presentation ¬
        result mouse activation mouse click)
set action of theAction to action type hyperlink action
set hyperlink address of hyperlink of theAction to "http://www.microsoft.com"
```

Remarks

If you've set properties of the **action setting** object that don't seem to be taking effect, make sure that you've set the *action* property to the appropriate value.

Properties

action

> Returns or sets the type of action that will occur when the specified shape is clicked or the mouse pointer is positioned over the shape during a slide show. Read/write.
>
> Can be one of the following constants:

action type end show	**action type next slide**
action type first slide	**action type none**
action type hyperlink action	**action type OLE verb**
action type last slide	**action type previous slide**
action type last slide viewed	**action type run macro**
action type unset	**action type run program**
action type named slide show	

You can use the **Action** property in conjunction with other properties of the **ActionSetting** object, as shown in the following table.

If you set the action property to this value	Use this property	To do this
action type hyperlink action	*hyperlink*	Set properties for the hyperlink that will be followed in response to a mouse action on the shape during a slide show.
action type run program	*action setting to run*	Return or set the name of the program to run in response to a mouse action on the shape during a slide show.
action type run macro	*action setting to run*	Return or set the name of the macro to run in response to a mouse action on the shape during a slide show.
action type OLE verb	*action verb*	Set the OLE verb that will be invoked in response to a mouse action on the shape during a slide show.
action type named slide show	*slide show name*	Set the name of the custom slide show that will run in response to a mouse action on the shape during a slide show.

action setting to run

Returns or sets the name of the presentation or macro to be run when the specified shape is clicked or the mouse pointer passes over the shape during a slide show. The *action* property must be set to **action type run macro** or **action type run program** for this property to affect the slide show action. Read/write.

Remarks

If the value of the *action* property is **action type run macro,** the specified string value should be the name of a global macro that's currently loaded. If the value of the *action* property is **action type run program**, the specified string value should be the full path and file name of a program.

You can set the *action setting to run* property to a macro that takes no arguments or a macro that takes a single **shape** or **object** argument. The shape that was clicked during the slide show will be passed as this argument.

action sound effect

Returns a **sound effect** object that represents the sound to be played when the specified shape is clicked or the mouse pointer passes over the shape. If you don't hear the sound that you assigned to the shape when you run the slide show, make sure that the *text level effect* property of the animation settings class is set to a value other than **animate level none** and that the *animate* property is set to **true**. Read-only.

action verb

> Returns or sets a string that contains the OLE verb that will be run when the user clicks the specified shape or passes the mouse pointer over it during a slide show. The *action* property must be set to **action type OLE verb** first for this property to affect the slide show action. Read/write.

animate action

> **True** if the color of the specified shape is momentarily inverted when the specified mouse action occurs. Read/write.

hyperlink

> Returns a **hyperlink** object that represents the hyperlink for the specified shape. For the hyperlink to be active during a slide show, the *action* property must be set to **action type hyperlink action**. Read-only.

slide show name

> Returns or sets the name of the custom slide show to run in response to a mouse action on the shape during a slide show. Read/write.

Class: add in

Plural

add ins

Represents a single add-in, either loaded or not loaded. The **add ins** list contains a list of all the add-ins available to PowerPoint, regardless of whether or not they're loaded. This list corresponds to the list of add-ins displayed in the **Add-Ins** dialog box (**Tools** menu).

Use **add in** *index*, where *index* is the title or index number of the add-in, to return a single **add in** object. The following example loads the My Ppt Tools add-in.

```
set loaded of add in "my ppt tools" to true
```

Don't confuse the add-in title, which appears in the **Add-Ins** dialog box, with the add-in name, which is the file name of the add-in. You must spell the add-in title exactly as it's spelled in the **Add-Ins** dialog box, but the capitalization doesn't have to match.

The index number represents the position of the add-in in the **Available add-ins** box in the **Add-Ins** dialog box. The following example displays the names of all the add-ins that are currently loaded in PowerPoint.

```
repeat with i from 1 to (get count of add ins)
    if loaded of add in i is true then display dialog (get name of add in i)
end repeat
```

Remarks

To add an add-in to the list of available add-ins, use the **register add in** command. Note, however, that using this method doesn't load the add-in. To load the add-in, set the *loaded* property of the add-in to **true** after you use the **register add in** command.

Use **presentation** *index*, where *index* is the file name (not the title) of the add-in, to return a reference to the presentation that corresponds to a loaded add-in. You must use the file name, because loaded add-ins usually don't appear in the **presentations** list. The following example sets the presAddin variable to the presentation for Myaddin.ppa.

```
set presAddin to presentation "myaddin.ppa"
```

Properties

auto load

> **True** if the specified add-in is automatically loaded each time PowerPoint is started. Read/write.

> Setting this property to **true** automatically sets the *registered* property to **true**.

full name

> Returns the name of the specified add-in , including the path, the current file system separator, and the file name extension. Read-only.

> This property is equivalent to the *path* property, followed by the current file system separator, followed by the *name* property.

loaded

> **True** if the specified add-in is loaded. In the **Add-Ins** dialog box (**Tools** menu), the check boxes next to loaded add-ins are selected. Read/write.

name

> The name of the add-in includes the file name extension (for file types that are registered) but doesn't include its path. Read-only.

path

> Returns the path to the specified **add in** object. Read-only.

> **Note**   The path doesn't include the final separator character or the name of the specified object. To return the file name without the path, use the *name* property of the **presentation** object. To return the file name and the path together, use the *full name* property.

registered

> **True** if the specified add-in is registered in the Preferences file. Read/write.

registered in HKLM

> **True** if the specified add-in is registered in the system registry. Read/write.

Class: animation behavior

Plural

animation behaviors

Represents the behavior of an animation effect, the main animation sequence, or an interactive animation sequence.

Use **behavior** *index*, where *index* is the number of the behavior in the sequence of behaviors, to return a single **animation behavior** object. The following example sets the positions of a rotation's starting and ending points. This example assumes that the first behavior for the main animation sequence is a **rotating effect** object.

```
set theAnimBhv to rotating effect of animation behavior 1 of main sequence ¬
        of timeline of slide 1 of active presentation
set rotating of theAnimBhv to 180
```

Properties

accumulate

Returns or sets whether animation behaviors accumulate. Read/write.

Can be one of the following:

- **no accumulate**
- **always**

additive

Returns or sets whether the current animation behavior is combined with other running animations. Read/write.

Can be one of the following:

- **no additive**
- **motion**

animation behavior type

> Returns or sets the type of animation. Read/write.
>
> Can be one of the following:

- **animation type none**
- **animation type motion**
- **animation type color**
- **animation type scale**
- **animation type rotation**
- **animation type property**
- **animation type command**
- **animation type filter**
- **animation type set**

colors effect

> Returns a **color effect** object that represents the color properties for a specified animation behavior. Read-only.

command effect

> Returns a **command effect** object for the specified animation behavior. Read-only.

filter effect

> Returns a **filter effect** object that represents a filter effect for the specified animation behavior. Read-only.

motion effect

> Returns a **motion effect** object that represents the properties of a motion animation. Read-only.

property effect

> Returns a **property effect** object for the specified animation behavior. Read-only.

rotating effect

> Returns a **rotating effect** object for the specified animation behavior. Read-only.

scale effect

> Returns a **scale effect** object for the specified animation behavior. Read-only.

set effect

> Returns a **set effect** object for the specified animation behavior. You can use the **set effect** object to set the value of a property. Read-only.

timing

> Returns a **timing** object that represents the timing properties for an animation sequence. Read-only.

Class: animation point

Plural

animation points

Represents an individual animation point for an animation behavior. The **animation points** list contains all the animation points for an animation behavior.

To add an **animation point** object, use the **make** command. Use **animation point** *index*, where *index* is the animation point's index number, to return a single **animation point** object. To set timing between animation points, use the *time* property of an **animation point** object. To set other animation point properties, such as color, use the *value* property. The following example adds three animation points to the first behavior in the active presentation's main animation sequence, and then the presentation changes colors at each animation point.

```
set sldNewSlide to make new slide at the beginning of active presentation ¬
        with properties {slide index:1, layout:slide layout blank}
set shpHeart to make new shape at sldNewSlide with properties ¬
        {auto shape type:autoshape heart, left position:100, top:100, ¬
        width:200, height:200}
set MSQ to main sequence of timeline of sldNewSlide
set effCustom to add effect MSQ for shpHeart fx animation type custom
set aniBehavior to add behavior effCustom type animation type property
tell property effect of anibehavior
    set property effect set to shape fill color
    repeat 3 times
        make new animation point at the beginning
    end repeat
    set time of animation point 1 to 0.2
    set value of animation point 1 to ({0, 0, 0} as RGB color)
    set time of animation point 2 to 0.5
    set value of animation point 2 to ({0, 255, 0} as RGB color)
    set time of animation point 3 to 1
    set value of animation point 3 to ({0, 255, 255} as RGB color)
end tell
```

Properties

formula

Returns or sets a Unicode text string that represents a formula to use for calculating an animation. Read/write.

time

Returns or sets a small real number that represents the time at a given animation point. Read/write.

value

Returns or sets a location reference that represents the value of a property for an animation point.

Class: animation settings

Represents the special effects applied to the animation for the specified shape during a slide show.

Use the *animation settings* property of a **shape** object to return the **animation settings** object. The following example adds a slide that contains both a title and a three-item list to the active presentation, and then it sets the list to be animated by first-level paragraphs, to fly in from the left when animated, to dim to the specified color after being animated, and to animate its items in reverse order.

```
set mySlide to make new slide at the end of slide 1 of active presentation ¬
        with properties {layout:slide layout text slide}
set content of text range of text frame of place holder 1 of mySlide ¬
        to "Top Three Reason"
set content of text range of text frame of place holder 2 of mySlide ¬
        to "Reason 1" & return & "Reason 2" & return & "Reason 3"
set text level effect of animation settings of place holder 2 of mySlide to ¬
        animate level first level
set entry effect of animation settings of place holder 2 of mySlide to ¬
        entry effect fly from left
set after effect of animation settings of place holder 2 of mySlide to ¬
        after effect dim
set dim color of animation settings of place holder 2 of mySlide to ¬
        {100, 120, 100} as RGB color
set animate text in reverse of animation settings of place holder 2 of ¬
        mySlide to true
```

Properties

advance time

> Returns or sets the amount of time after which the specified shape will become animated. Read/write.

after effect

> Returns or sets a value that indicates whether the specified shape appears dimmed, hidden, or unchanged after it's been built. Read/write.
>
> Can be one of the following:

- **after effect dim**
- **after effect hide**
- **after effect unset**
- **after effect none**

> The default value is **after effect none**.
>
> You won't see the after effect you set for a shape unless the shape gets animated and at least one other shape on the slide gets animated after it. For a shape to be animated, the *text level effect* property must be set to something other than **animate level none** and the *animate* property must be set to **true**. To change the build order of the shapes on a slide, use the *animation order* property.

animate

> **True** if the specified shape is animated during a slide show. Read/write.
>
> This property must be set to **true** for any of the other properties of the **animation settings** object to take effect.

animate background

> If the specified object is an AutoShape, **true** if the shape is animated separately from the text it contains. If the specified shape is a chart object, **true** if the background (the axes and gridlines) of the specified graph object is animated. Applies only to AutoShapes with text that can be built in more than one step or to graph objects. Read/write.
>
> To control the animation of text attached to the specified shape, use the *text level effect* and *text unit effect* properties.
>
> If this property is set to **true** and the *text level effect* property is set to **animate level all levels**, the shape and its text will be animated simultaneously. If this property is set to **true** and the *text level effect* property is set to anything other than **animate level all levels**, the shape will be animated immediately before the text is animated.
>
> You won't see effects of setting this property unless the specified shape is animated. For a shape to be animated, the *text level effect* property must be set to something other than **animate level none** and the *animate* property must be set to **true**.

animate text in reverse

> **True** if the specified shape is built in reverse order. Applies only to shapes (such as shapes containing lists) that can be built in more than one step. Read/write.
>
> You won't see effects of setting this property unless the specified shape gets animated. For a shape to be animated, the *text level effect* property of the **animation settings** object for the shape must be set to something other than **animate level none** and the *animate* property must be set to **true**.

animation order

> Returns or sets an integer that represents the position of the specified shape within the collection of shapes to be animated. Read/write.
>
> You won't see effects of setting this property unless the specified shape gets animated. For a shape to be animated, the *text level effect* property of the **animation settings** object for the shape must be set to something other than **animate level none** and the *animate* property must be set to **true**.

animation play settings

> Returns a **play settings** object that contains information about how the specified media clip plays during a slide show. Read-only.

animation sound effect

> Returns a **sound effect** object that represents the sound to be played during the animation of the specified shape. Read-only.

chart unit effect

> Returns or sets a value that indicates whether the graph range is animated by series, category, or element. Read/write.
>
> Can be one of the following:

- **chart unit effect by category**
- **chart unit effect by series element**
- **chart unit effect by series**
- **animation chart unset**

> If your graph doesn't become animated, make sure that the *animate* property is set to **true**.

dim color

> Returns or sets an RGB color that represents the color of the specified shape after it's been built. Read-only.
>
> If you don't get the effect you expect, check your other build settings. You won't see the effect of the *dim color* property unless the *text level effect* property is set to something other than **animate level none**, the *after effect* property is set to **after effect dim**, and the *animate* property is set to **true**. In addition, if the specified shape is the only item or the last item to be built on the slide, the shape won't be dimmed. To change the build order of the shapes on a slide, use the *animation order* property.

entry effect

Returns or sets the special effect applied to the animation for the specified shape. Read/write. Can be one of the following:

entry effect appear	**entry effect unset**
entry effect blinds horizontal	**entry effect none**
entry effect blinds vertical	**entry effect peek from down**
entry effect box in	**entry effect peek from left**
entry effect box out	**entry effect peek from right**
entry effect checkerboard across	**entry effect peek from up**
entry effect checkerboard down	**entry effect plus**
entry effect circle	**entry effect random**
entry effect collapse across	**entry effect random bars horizontal**
entry effect collapse left	**entry effect random bars vertical**
entry effect collapse up	**entry effect spinner**
entry effect collapse right	**entry effect spiral**
entry effect collapse bottom	**entry effect split horizontal in**
entry effect cover down	**entry effect split horizontal out**
entry effect cover left	**entry effect split vertical in**
entry effect cover left down	**entry effect split vertical out**
entry effect cover left up	**entry effect strips down left**
entry effect cover right	**entry effect strips down right**
entry effect cover right down	**entry effect strips left down**
entry effect cover right up	**entry effect strips left up**
entry effect cover up	**entry effect strips right down**
entry effect crawl from down	**entry effect strips right up**
entry effect crawl from left	**entry effect strips up left**
entry effect crawl from right	**entry effect strips up right**
entry effect crawl from up	**entry effect swivel**
entry effect cut	**entry effect uncover down**
entry effect diamond	**entry effect uncover left**
entry effect dissolve	**entry effect uncover left down**
entry effect fade	**entry effect uncover left up**
entry effect fade smoothly	**entry effect uncover right**
entry effect fade fly from left	**entry effect uncover right down**
entry effect fade fly from top	**entry effect uncover right up**
entry effect fade fly from right	**entry effect uncover up**
entry effect fade fly from bottom	**entry effect wedge**
entry effect fade fly from top left	**entry effect wheel1 spoke**
entry effect fade fly from top right	**entry effect wheel2 spokes**
entry effect fade fly from bottom left	**entry effect wheel3 spokes**
entry effect fade fly from bottom right	**entry effect wheel4 spokes**
entry effect flash once fast	**entry effect wheel8 spokes**
entry effect flash once medium	**entry effect wipe down**
entry effect flash once slow	**entry effect wipe left**
entry effect fly from bottom	**entry effect wipe right**
entry effect fly from bottom left	**entry effect wipe up**
entry effect fly from bottom right	**entry effect zoom in**
entry effect fly from left	**entry effect zoom in slightly**
entry effect fly from right	**entry effect zoom out**
entry effect fly from top	**entry effect zoom out slightly**
entry effect fly from top left	**entry effect zoom center**
entry effect fly from top right	**entry effect zoom bottom**

If the *text level effect* property for the specified shape is set to **animate level none** (the default value) or the *animate* property is set to **false**, you won't see the special effect you've applied with the *entry effect* property.

text level effect

Returns or sets a value that indicates whether the text in the specified shape is animated by first-level paragraphs, second-level paragraphs, or some other level (up to fifth-level paragraphs). Read/write.

Can be one of the following:

- **animate level all levels**

- **animate level fifth level**

- **animate level first level**

- **animate level fourth level**

- **animate level second level**

- **animate level third level**

- **animation level unset**

- **animate level none**

For the *text level effect* property setting to take effect, the *animate* property must be set to **true**.

text unit effect

Returns or sets a value that indicates whether the text in the specified shape is animated paragraph by paragraph, word by word, or letter by letter. Read/write.

Can be one of the following:

- **text unit effect by character**

- **text unit effect by paragraph**

- **text unit effect by word**

- **animation unit unset**

For the *text unit effect* property setting to take effect, the *text level effect* property for the specified shape must have a value other than **animate level none** or **animate level all levels**, and the *animate* property must be set to **true**.

Class: application

Elements

presentation

document window

slide show window

command bar

add in

Represents the entire Microsoft PowerPoint application. The **application** object contains:

- Application-wide settings and options (the name of the active printer, for example).

- Properties that return top-level objects, such as **active presentation**, **document windows**, and so on.

Use **application** "Microsoft PowerPoint" to return the **application** object.

The following example launches PowerPoint and opens an existing presentation named "Ex_a2a.ppt".

```
tell application "Microsoft PowerPoint"
    launch
    set visible to true
    open "Macintosh HD:Users:Shared:Ex_a2a.ppt"
end tell
```

Properties

Version

Returns the PowerPoint version number. Read-only.

active presentation

Returns a **presentation** object that represents the presentation open in the active window. Read-only.

Note that if an embedded presentation is in-place active, the *active presentation* property returns the embedded presentation.

active printer

Returns the name of the active printer. Read-only.

active window

Returns a **document window** object that represents the active document window. Read-only.

build

Returns the PowerPoint build number. Read-only.

caption

Returns the text that appears in the title bar of the application window. Read-write.

default save format

Returns or sets the default save format. Read/write.

Can be one of the following:

- **save as presentation**
- **save as template**
- **save as RTF**
- **save as show**
- **save as addIn**
- **save as default**
- **save as HTML**
- **save as movie**

default web options object

Returns the **default web options** object, which contains global application-level attributes used by PowerPoint when you publish or save a presentation as a Web page or open a Web page. Read-only.

name

Returns the string "Microsoft PowerPoint." Read-only.

office assistant

Returns an **assistant** object that represents the Office Assistant. Read-only.

operating system

Returns the name of the operating system. Read-only.

path

Returns the path to the specified **application** object. Read-only.

Note   The path doesn't include the final separator character or the name of the specified object.

save as movie settings object

> Returns the **save as movie settings** object. When you derive a **save as movie settings** object using the *save as movie settings object* property, any changes that you make to the properties of the **save as movie settings** object affect the specific presentation, not the default preferences for the application.

start up dialog

> True if the Project Gallery is displayed at startup. Read/write.

Class: bullet format

Represents bullet formatting.

To return the **bullet format** object, use the *bullet format* property of the **paragraph format** class. The following example sets the bullet size and color for the paragraphs in shape two on slide one in the active presentation.

```
set myShape to shape 2 of slide 1 of active presentation
set myBullet to bullet format of paragraph format of text range ¬
        of text frame of myShape
set relative size of myBullet to 1.25
set bullet character of myBullet to "169"
set font color of font of myBullet to {255, 255, 0} as RGB color
set font name of font of myBullet to "Symbol"
```

Properties

bullet character

> Returns or sets the Unicode character that is used for bullets in the specified text. Read/write.

bullet font

> Returns a **font** object that represents the bullet formatting. Read-only.

bullet number

> Returns the bullet number of a paragraph when the *bullet type* property is set to **bullet type numbered**. Read-only.

bullet start value

> Returns or sets the beginning value of a bulleted list when the *bullet type* property is set to **bullet type numbered**. The value must be in the range of 1 to 32767. Read/write.

bullet style

Returns or sets the style of the bullet.

Can be one of the following:

numbered bullet style alpha lowercase period
numbered bullet style alpha uppercase period
numbered bullet style arabic right paren
numbered bullet style arabic period
numbered bullet style roman lowercase paren both
numbered bullet style roman lowercase paren right
numbered bullet style roman lowercase period
numbered bullet style roman uppercase period
numbered bullet style alpha lowercase paren both
numbered bullet style alpha lowercase paren right
numbered bullet style alpha uppercase paren both
numbered bullet style alpha uppercase paren right
numbered bullet style arabic paren both
numbered bullet style arabic plain
numbered bullet style roman uppercase paren both
numbered bullet style roman uppercase paren right
numbered bullet style simplified chinese plain
numbered bullet style simplified chinese period
numbered bullet style circle number plain
numbered bullet style circle number white plain
numbered bullet style circle number black plain
numbered bullet style traditional chinese plain
numbered bullet style traditional chinese period
numbered bullet style arabic alpha dash
numbered bullet style arabic abjad dash
numbered bullet style hebrew alpha dash
numbered bullet style kanji korean plain
numbered bullet style kanji korean period
numbered bullet style arabic DB plain
numbered bullet style unset

Some of these constants may not be available to you, depending on the language support (U.S. English, for example) that you've selected or installed. Read/write.

bullet type

Returns or sets the bullet type. Read/write.

Can be one of the following:

- **bullet type unset**

- **bullet type none**

- **bullet type unnumbered**

- **bullet type numbered**

- **picture bullet type**

relative size

> Returns or sets the bullet size relative to the size of the first text character in the paragraph. Can be a floating-point value from 0.25 through 4, indicating that the bullet size can be from 25 percent through 400 percent of the text-character size. Read/write.

use text color

> **True** if the specified bullets are set to the color of the first text character in the paragraph. **False** if the specified bullets are set to any other color. Read/write.

> You cannot explicitly set this property to **false**. Setting the bullet format color (using the *font color* property of the **font** object) sets this property to **false**. When *use text color* is **false**, you can set it to **true** to reset the bullet format to the default color.

use text font

> **True** if the specified bullets are set to the font of the first text character in the paragraph. **False** if the specified bullets are set to a custom font. Read/write.

> You cannot explicitly set this property to **false**. Setting the bullet format font (using the *font name* property of the **font** object) sets this property to **false**. When *use text font* is **false**, you can set it to **true** to reset the bullet format to the default font.

visible

> **True** if the specified object, or the formatting applied to the specified object, is visible. Read/write.

Class: color scheme

Plural

color schemes

Represents a color scheme, which is a set of eight colors used for the different elements of a slide, notes page, or handout, such as the title or background. (Note that the color schemes for slides, notes pages, and handouts in a presentation can be set independently.) Each color is represented by an RGB color. The **color schemes** list contains all the color schemes in a presentation.

This section describes how to do the following:

- Return a **color scheme** object from the list of all the color schemes in the presentation

- Return the **color scheme** object attached to a specific slide or master

- Return the color of a single slide element from a **color scheme** object

Returning a color scheme object from the list of all the color schemes in the presentation

Use **color scheme** *index*, where *index* is the color scheme index number, to return a single **color scheme** object. The following example deletes color scheme two from the active presentation.

```
delete color scheme 2 of active presentation
```

Returning the color scheme object attached to a specific slide or master

Set the *color scheme* property of a **slide** or **master** object to return the color scheme for one slide or a master, respectively. The following example changes the color scheme for the current slide.

```
set theScheme to (get color scheme of slide of view of active window)
set color for theScheme at title scheme to color ({0, 150, 250} as RGB color)
```

Returning the color of a single slide element from a color scheme object

Use the **get color from** command to return an **RGB color** object that represents the color of a single slide-element type. Use the **set color for** command to set the color of a single slide-element type. The following example sets the title color of color scheme 2 to the title color that's defined for color scheme one.

```
set theTitle to (get color from color scheme 1 of active presentation ¬
        at title scheme)
set color for color scheme 2 of active presentation at title scheme ¬
        to color theTitle
```

Class: colors effect

Represents a color effect for an animation behavior.

Use the *colors effect* property of the **animation behavior** object to return a **colors effect** object. Color effects can be returned using the **colors effect** object's *color* property. You cannot change the **colors effect** object for an animation behavior programmatically.

Properties

color

> Returns an RGB value that represents the color of the object.

Class: command effect

Represents a command effect for an animation behavior. You can send events, call functions, and OLE verbs to embedded objects using this object.

To return a **command effect** object, use the *command effect* property of the **animation behavior** object. To change command effects, use the *command* and *type* properties of the **command effect** object.

The following example shows how to set a command effect animation behavior for an effect named effectNew.

```
set bhvEffect to add behavior effectNew type animation type command
set type of command effect of bhvEffect to verb
set command of command effect of bhvEffect to "Play"
```

Properties

command

> Returns or sets a string that represents the command to be executed for the command effect. Read/write.

type

> Returns or sets the type of command to be executed for the command effect. Read/write.
>
> Can be one of the following:

- **event**
- **call**
- **verb**

Class: default web options

Contains global application-level attributes used by PowerPoint when you publish or save a presentation as a Web page or when you open a Web page. You can return or set attributes either at the application (global) level or at the presentation level. (Note that attribute values can be different from one presentation to another, depending on the attribute value at the time the presentation was saved.) Presentation-level attribute settings override application-level attribute settings. Presentation-level attributes are contained in the **web options** object.

Using the default web options Object

To return the **default web options** object, use the *default web options object* property of the application class. The following example checks to see whether PNG (Portable Network Graphics) are allowed as an image format, and then sets the `strImageFileType` variable accordingly.

```
set objAppWebOptions to default web options object
if allow PNG of objAppWebOptions is true then
    set strImageFileType to "PNG"
else
    set strImageFileType to "JPG"
end if
```

Properties

allow PNG

> **True** if PNG (Portable Network Graphics) is allowed as an image format when you save or publish a presentation as a Web page. **False** if PNG is not allowed as an output format. The default value is **false**. Read/write.
>
> If you save images in the PNG format as opposed to any other file format, you might improve the image quality or reduce the size of those image files, and therefore decrease the download time, assuming that the Web browsers you are targeting support the PNG format.

always save in default encoding

> **True** if the default encoding is used when you save a Web page or plain text document, independent of the file's original encoding when opened. **False** if the original encoding of the file is used. The default value is **false**. Read/write.

> The *encoding* property can be used to set the default encoding.

buttons type

> Determines the type of navigation buttons PowerPoint uses in a Web version of a presentation. Read/write.

> Can be one of the following:

> - **fancy** (graphic buttons)
> - **regular**
> - **text only**

check if Office is HTML editor

> **True** if PowerPoint checks to see whether an Office application is the default HTML editor when you start PowerPoint. **False** if PowerPoint does not perform this check. The default value is **true**. Read/write.

> This property is used only if the Web browser you are using supports HTML editing and HTML editors.

> To use a different HTML editor, you must set this property to **false** and then register the editor as the default system HTML editor.

encoding

> Returns or sets the document encoding (code page or character set) to be used by the Web browser when you view the saved document.

> Can be one of the following:

encoding Thai	**encoding EBCDIC Greek**
encoding Japanese ShiftJIS	**encoding EBCDIC Hebrew**
encoding simplified Chinese	**encoding EBCDIC Korean extended**
encoding Korean	**encoding EBCDIC Thai**
encoding traditional Chinese	**encoding EBCDIC Icelandic**
encoding little endian	**encoding EBCDIC Turkish**
encoding big endian	**encoding EBCDIC Russian**
encoding central European	**encoding EBCDIC Serbian Bulgarian**
encoding Cyrillic	**encoding EBCDIC Japanese Katakana**
encoding Western	**extended and Japanese**
encoding Greek	**encoding EBCDIC US Canada and Japanese**
encoding Turkish	**encoding EBCDIC extended and Korean**
encoding Hebrew	**encoding EBCDIC simplified Chinese**
encoding Arabic	**extended and simplified Chinese**
encoding Baltic	**encoding EBCDIC US Canada**
encoding Vietnamese	**and traditional Chinese**
encoding auto detect	**encoding EBCDIC Japanese Latin extended**
encoding Japanese auto detect	**and Japanese**
encoding simplified Chinese auto detect	**encoding OEM United States**

encoding Korean auto detect
encoding traditional Chinese auto detect
encoding Cyrillic auto detect
encoding Greek auto detect
encoding Arabic auto detect
encoding ISO88591 Latin1
encoding ISO88592 central Europe
encoding ISO88593 Latin3
encoding ISO88594 Baltic
encoding ISO88595 Cyrillic
encoding ISO88596 Arabic
encoding ISO88597 Greek
encoding ISO88598 Hebrew
encoding ISO88599 Turkish
encoding ISO885915 Latin9
encoding ISO2022 Japanese no half width
   Katakana
encoding ISO2022 Japanese JISX02021984
encoding ISO2022 Japanese JISX02011989
encoding ISO2022KR
encoding ISO2022CN traditional Chinese
encoding ISO2022CN simplified Chinese
encoding Mac Roman
encoding Mac Japanese
encoding Mac traditional Chinese Big5
encoding Mac Korean
encoding Mac Greek1
encoding Mac Cyrillic
encoding Mac simplified Chinese GB2312
encoding Mac Romania
encoding Mac Ukraine
encoding Mac Latin2
encoding Mac Icelandic
encoding Mac Turkish
encoding Mac Croatia
encoding EBCDIC US Canada
encoding EBCDIC International
encoding EBCDIC multilingual
   ROECE Latin2
encoding EBCDIC Greek modern
encoding EBCDIC Turkish Latin5
encoding EBCDIC Germany
encoding EBCDIC Denmark Norway
encoding EBCDIC Finland Sweden
encoding EBCDIC Italy
encoding EBCDIC Latin America Spain
encoding EBCDIC United Kingdom
encoding EBCDIC Japanese
   Katakana extended
encoding EBCDIC France
encoding EBCDIC Arabic

encoding OEM Greek
encoding OEM Baltic
encoding OEM multilingual LatinI
encoding OEM multilingual LatinII
encoding OEM Cyrillic
encoding OEM Turkish
encoding OEM Portuguese
encoding OEM Icelandic
encoding OEM Hebrew
encoding OEM Canadian French
encoding OEM Arabic
encoding OEM Nordic
encoding OEM CyrillicII
encoding OEM modern Greek
encoding EUC Japanese
encoding EUC Chinese simplified Chinese
encoding EUC Korean
encoding EUC Taiwanese
   traditional Chinese
encoding Devanagari
encoding Bengali
encoding Tamil
encoding Telugu
encoding Assamese
encoding Oriya
encoding Kannada
encoding Malayalam
encoding Gujarati
encoding Punjabi
encoding Arabic ASMO
encoding Arabic transparent ASMO
encoding Korean Johab
encoding Taiwan CNS
encoding Taiwan TCA
encoding Taiwan Eten
encoding Taiwan IBM5550
encoding Taiwan teletext
encoding Taiwan Wang
encoding IA5 German
encoding IA5 Swedish
encoding IA5 Norwegian
encoding US ASCII
encoding T61
encoding ISO6937 nonspacing accent
encoding Ext alpha lowercase
encoding KOI8U
encoding Europa3
encoding HZGB simplified Chinese
encoding UTF7
encoding UTF8

The default is the system code page. Read/write.

frame colors

> Returns or sets the text color for the notes pane and the outline pane, and the background color for the notes pane, outline pane, and slide pane for Web presentations. Read/write.
>
> Can be one of the following:

- **frame colors browser colors**
- **frame colors presentation scheme text color**
- **frame colors presentation scheme accent color**
- **frame colors white text on black**
- **frame colors black text on white**

include binary file

> Specifies whether to include a binary version of the presentation with the Web version. Including a binary version makes it possible to open and edit the Web version in PowerPoint. Read/write.

nav bar placement

> Determines the position of the navigation bar. Read/write.
>
> Can be one of the following:

- **bar placement bottom**
- **bar placement top**

support IE4

> Determines whether a Web presentation is optimized for Microsoft Internet Explorer 4.0 or later. Read/write.

support NN4

> Determines whether a Web presentation is optimized for Netscape Navigator 4.0 or later. Read/write.

support older browsers

> Determines whether a Web presentation is optimized for browsers that are older than Netscape Navigator 4.0 or Microsoft Internet Explorer 4.0. Read/write.

update links on save

> **True** if hyperlinks and paths to all supporting files are automatically updated before you save or publish the presentation as a Web page, ensuring that the links are up-to-date at the time the presentation is saved. **False** if the links are not updated. The default value is **true**. Read/write.
>
> You should set this property to **false** if the location where the presentation is saved is different from the final location on the Web server and the supporting files are not available at the first location.

web page keywords

> Returns or sets keywords for a presentation when it is saved as a Web page. Read/write.

> Keywords are not visible to Web-page readers but are stored in meta tags to facilitate page searches.

web page title

> Sets the title of a Web presentation. Read/write.

> This title will appear as the title of the browser window in which the Web presentation is viewed.

Class: document window

Plural

document windows

Elements

pane

Represents a document window. The **document windows** list contains all the open document windows.

Use **document window** *index*, where *index* is the document window index number, to return a single **document window** object. The following example activates document window two.

```
activate document window 2
```

To return the presentation that's currently running in the specified document window, use the *presentation* property. To return the view in the specified document window, use the *view* property.

Properties

active

> **True** if the specified window is the active window. Read-only.

active pane

> Returns a **pane** object that represents the active pane in the document window. Read-only.

black and white

> **True** if the document window display is black and white. The default value is **false**. Read/write.

caption

> Returns the text that appears in the title bar of the document window. Read-only.

entry index

> Returns a number that indicates the position of the document window in the document windows list. Read-only.

height

> Returns or sets the height (in points) of the specified window. Read/write.

left position

> Returns or sets the distance from the left edge of the document window to the left edge of the application window's client area. Read/write.

presentation

> Returns a **presentation** object that represents the presentation in which the specified document window or slide show window was created. Read-only.

split horizontal

> Returns or sets the percentage of the document window width that the outline pane occupies in normal view. Corresponds to the pane divider position between the slide and outline panes. Read/write.

> The maximum value of the *split horizontal* property is always less than 100% because the slide pane has a minimum width that depends on a 10% zoom level. The actual maximum value may vary depending on the size of the application window.

split vertical

> Returns or sets the percentage of the document window height that the slide pane occupies in normal view. Corresponds to the pane divider position between the slide and notes panes. Read/write.

> The minimum value of the *split vertical* property is always greater than 0% because the slide pane has a minimum height that depends on a 10% zoom level. The actual minimum value may vary depending on the size of the application window.

top

> Returns or sets the distance from the top edge of the document window to the top edge of the application window's client area. Read/write.

view

> Returns a **view** object that represents the view in the specified document window. Read-only.

view type

> Returns or sets the type of the view contained in the specified document window. Read/write.

> Can be one of the following:

> - **normal view**
> - **handout master view**
> - **notes master viewv**
> - **outline view**
> - **page view**
> - **master view**
> - **slide view**
> - **slide sorter view**
> - **title master view**

width

> Returns or sets the width (in points) of the specified object. Read/write.

Class: effect

Plural

effects

Elements

animation behavior

Represents timing information about a slide animation.

To add an effect, use the **add effect** command. This example adds a shape to the first slide in the active presentation and adds an effect and a behavior to the shape.

```
set sldOne to slide 1 of active presentation
set shpStar to make new shape at the beginning of sldOne with properties ¬
        {auto shape type:autoshape five point star, left position:150, ¬
        top:72, width:400, height:400}
set effNew to add effect (main sequence of timeline of sldOne) for shpStar ¬
        fx animation type stretchy trigger after previous
set effBHV to add behavior effNew type animation type scale
set from x of scale effect of effBHV to 75
set from y of scale effect of effBHV to 75
set to x of scale effect of effBHV to 0
set to y of scale effect of effBHV to 0
set autoreverse of timing of effNew to true
```

To refer to an existing **effect** object, use **effect** *index*, where *index* is the number of the **effect** object in the **sequence** list. This example changes the effect for the main sequence.

```
set MSQ to main sequence of timeline of slide 1 of active presentation
set animation effect type of effect 1 of MSQ to animation type spin
```

There is always at least one **effect** object in each slide regardless of whether or not the slide contains animations.

Properties

animation effect type

Returns or sets the animation effect type. Read/write.

Can be one of the following:

animation type custom	**animation type flash bulb**
animation type appear	**animation type flicker**
animation type fly	**animation type grow with color**
animation type blinds	**animation type lighten**
animation type box	**animation type style emphasis**
animation type checkerboard	**animation type teeter**
animation type circle	**animation type vertical grow**
animation type crawl	**animation type wave**
animation type diamond	**animation type media play**
animation type dissolve	**animation type media pause**
animation type fade	**animation type media stop**
animation type flash once	**animation type circle path**
animation type peek	**animation type right triangle path**
animation type plus	**animation type diamond path**
animation type random bars	**animation type hexagon path**
animation type spiral	**animation type 5 point star path**
animation type split	**animation type crescent moon path**
animation type stretch	**animation type square path**
animation type strips	**animation type trapezoid path**
animation type swivel	**animation type heart path**
animation type wedge	**animation type octagon path**
animation type wheel	**animation type 6 point star path**
animation type wipe	**animation type football path**
animation type zoom	**animation type equal triangle path**
animation type random effect	**animation type parallelogram path**
animation type boomerang	**animation type pentagon path**
animation type bounce	**animation type 4 point star path**
animation type color reveal	**animation type 8 point star path**
animation type credits	**animation type teardrop path**
animation type ease in	**animation type pointy star path**
animation type float	**animation type curved square path**
animation type grow and turn	**animation type curved x path**
animation type light speed	**animation type vertical figure 8 path**
animation type pinwheel	**animation type curvy star path**
animation type rise up	**animation type loop de loop path**
animation type swish	**animation type buzzsaw path**
animation type thin line	**animation type horizontal figure 8 path**
animation type unfold	**animation type peanut path**
animation type whip	**animation type figure 8 four path**
animation type ascend	**animation type neutron path**
animation type center revolve	**animation type swoosh path**
animation type faded swivel	**animation type bean path**
animation type descend	**animation type plus path**
animation type sling	**animation type inverted triangle path**
animation type spinner	**animation type inverted square path**
animation type stretchy	**animation type left path**

animation type zip	**animation type turn right path**
animation type arc up	**animation type arc down path**
animation type fade zoom	**animation type zigzag path**
animation type glide	**animation type s curve 2 path**
animation type expand	**animation type sine wave path**
animation type flip	**animation type bounce left path**
animation type shimmer	**animation type down path**
animation type fold	**animation type turn up path**
animation type change fill color	**animation type arc up path**
animation type change font	**animation type heartbeat path**
animation type change font color	**animation type spiral right path**
animation type change font size	**animation type wave path**
animation type change font style	**animation type curvy left path**
animation type grow shrink	**animation type diagonal down right path**
animation type change line color	**animation type turn down path**
animation type spin	**animation type arc left path**
animation type transparency	**animation type funnel path**
animation type bold flash	**animation type spring path**
animation type blast	**animation type bounce right path**
animation type bold reveal	**animation type spiral left path**
animation type brush on color	**animation type diagonal up right path**
animation type brush on underline	**animation type turn up right path**
animation type color blend	**animation type arc right path**
animation type color wave	**animation type s curve 1 path**
animation type complementary color	**animation type decaying wave path**
animation type complementary color 2	**animation type curvy right path**
animation type contrasting color	**animation type stairs down path**
animation type darken	**animation type up path**
animation type desaturate	**animation type right path**

effect information

Returns an **effect information** object representing information for a specified animation effect. Read-only.

effect parameters

Returns an **effect parameters** object representing animation effect properties. Read-only.

exit animation

Returns or sets whether the animation effect is an exit effect. Read/write.

name

Returns a string that represents the name of an animation effect. Read-only.

paragraph

Returns or sets an integer that represents the paragraph in a text range to which to apply animation effects. Read/write.

shape

Returns a **shape** object that represents an animated shape. Read-only.

text range length

>   Returns or sets an integer that represents the length of a text range. Read-only.

text range start

>   Returns or sets an integer that represents the start of a text range. Read-only.

timing

>   Returns a **timing** object that represents the timing properties for an animation sequence. Read-only.

Class: effect information

Represents various animation options for an **effect** object.

Use the **effect information** class to return the current state of an **effect** object, such as the after effect, whether the background animates along with its corresponding text, whether text animates in reverse, play settings, sound effects, text building behavior, and so on. All of the properties of the **effect information** object are read-only. To change any effect information properties, you must use the commands and properties of the corresponding **effect** object.

To return an **effect information** object, use the *effect information* property of the **effect** object. The following example sets the *hide while not playing* property for the play settings in the main animation sequence.

```
set playSet to play settings information of effect information of ¬
        main sequence of timeline of slide 1 of active window
set hide while not playing of playSet to true
```

Properties

after effect information

>   Returns whether an after effect is dimmed, hidden, or unchanged after it runs. Read-only.

>   Can be one of the following:

>   - **no after effect**
>   - **dim**
>   - **hide**
>   - **hide on next click**

animate background information

>   Returns **true** if the specified effect is a background animation. Read-only.

animate text in reverse information

> **True** if the specified shape is built in reverse order. Applies only to shapes (such as shapes containing lists) that can be built in more than one step. Read-only.
>
> You won't see the effects of setting this property unless the specified shape gets animated. For a shape to be animated, the *text level effect* property of the **animation settings** object for the shape must be set to something other than **animate level none** and the *animate* property must be set to **true**.

build by level

> Returns the level of the animation build effect. Read-only.
>
> Can be one of the following:

chart all at once	**text by all levels**
chart by category	**text by fifth level**
chart by category elements	**text by first level**
chart by series	**text by fourth level**
chart by series elements	**text by second level**
text by no levels	**text by third level**

dim

> Returns an RGB color that represents the color to dim to after an animation is finished. Read only

play settings information

> Returns a **play settings** object that contains information about how the specified media clip plays during a slide show. Read-only.

sound effect information

> Returns a **sound effect** object that represents the sound to be played during the animation of the specified shape. Read-only.

text unit effect information

> Returns whether the text in the specified shape is animated paragraph by paragraph, word by word, or letter by letter. Read-only.
>
> Can be one of the following:

- **by paragraph**
- **by character**
- **by word**

Class: effect parameters

Represents various animation parameters for an **effect** object, such as colors, fonts, sizes, and directions.

To return an **effect parameters** object, use the *effect parameters* property of the **effect** object. The following example creates a shape, sets a spin effect, and changes the spin direction.

```
set shpNew to make new shape at the beginning of slide 1 of ¬
        active presentation with properties {left position:100, top:100, ¬
        width:150, height:150, auto shape type:autoshape heart}
set effNew to add effect (main sequence of timeline of slide 1 of ¬
        active presentation) for shpNew fx animation type spin
set direction of effect parameters of effNew to up left
```

Properties

amount

>   Returns or sets a small real number that represents the number of degrees an animated shape is rotated around the z-axis. A positive value indicates clockwise rotation; a negative value indicates counterclockwise rotation. Read/write.

color2

>   Returns an RGB color that represents the color on which to end a color-cycle animation. Read-only.

direction

>   Returns or sets the direction used for an animation effect. This property can be used only if the effect uses a direction. Read/write.

>   Can be one of the following:

across	**font shadow**	**out**
bottom	**font strikethrough**	**out bottom**
bottom left	**font underline**	**out center**
bottom right	**gradual**	**out slightly**
center	**horizontal**	**right**
clockwise	**horizontal in**	**slightly**
counterclockwise	**horizontal out**	**top**
cycle clockwise	**in**	**top left**
cycle counterclockwise	**in bottom**	**top right**
down	**in center**	**up**
down left	**in slightly**	**up left**
down right	**instant**	**up right**
font all caps	**left**	**vertical**
font bold	**no direction**	**vertical in**
font italic	**ordinal mask**	**vertical out**

font

>   Returns or sets the name of the font in the specified effect. Read/write.

relative

> **True** to set the motion position relative to the position of the shape. This property is used only in conjunction with motion paths. Read/write.

size

> Returns or sets the character size, in points. Read/write.

Class: filter effect

Represents a filter effect for an animation behavior.

To return a **filter effect** object, use the *filter effect* property of the **animation behavior** object. To change filter effects, use the *reveal*, *subtype*, and *filter type* properties of the **filter effect** object.

The following example adds a shape to the first slide of the active presentation and sets a filter effect animation behavior.

```
set sldFirst to slide 1 of active presentation
set shpHeart to make new shape at the beginning of sldFirst with properties ¬
        {auto shape type:autoshape heart, left position:100, top:100, ¬
        width:100, height:100}
set effNew to add effect (main sequence of timeline of sldFirst) ¬
        for shpHeart fx animation type change fill color ¬
        trigger after previous
set bhvEffect to add behavior effNew type animation type filter
set filter type of bhvEffect to wipe
set subtype of filter effect of bhvEffect to subtype up
set reveal of filter effect of bhvEffect to true
```

Properties

filter type

> Returns or sets the type of filter. Read/write.
>
> Can be one of the following:

no effect type	**pixelate**
barn	**plus**
blinds	**random bar**
box	**slide**
checkerboard	**stretch**
circle	**strips**
diamond	**wedge**
dissolve	**wheel**
fade	**wipe**
image	

reveal

> Determines how the embedded objects will be revealed. Read/write.

> Setting a value of **true** for the *reveal* property when the filter effect type is **wipe** will make the shape appear. Setting a value of **false** will make the object disappear. In other words, if your filter is set to wipe and *reveal* is **true**, you will get a wipe in effect; when *reveal* is **false**, you will get a wipe out effect.

subtype

> Returns or sets the subtype of the filter effect. Read/write.

> Can be one of the following:

no effect subtype	**down left**
in vertical	**up left**
out vertical	**down right**
in horizontal	**up right**
out horizontal	**spoke 1**
horizontal	**spokes 2**
vertical	**spokes 3**
out	**spokes 4**
across	**spokes 8**
from left	**subtype left**
from right	**subtype right**
from top	**subtype down**
from bottom	**subtype up**

Class: font

Represents character formatting for text or a bullet.

Use the *font* property to return the **font** object that represents the font attributes for a specific bullet, text range, or outline level. The following example sets the title text on slide one and sets the font properties.

```
set sldTitle to shape 1 of slide 1 of active presentation

set content of text range of text frame of sldTitle to "Volcano Coffee"

set italic of font of text range of text frame of sldTitle to true

set font name of font of text range of text frame of sldTitle to "Palatino"

set font color of font of text range of text frame of sldTitle to ¬
        ({0, 0, 255} as RGB color)
```

Properties

ASCII name

> Returns or sets the font used for ASCII characters (characters with character set numbers within the range of 0 to 27). Read/write.

auto rotate numbers

> Returns or sets lateral compression. Read/write.

base line offset

> Returns or sets the base line offset for the specified superscript or subscript characters. Can be a floating-point value from -1 through 1. A value of -1 represents an offset of -100 percent, and a value of 1 represents an offset of 100 percent. Read/write.

> Setting the *base line offset* property to a negative value automatically sets the *subscript* property to **true** and the *superscript* property to **false**.

> Setting the *base line offset* property to a positive value automatically sets the *subscript* property to **false** and the *superscript* property to **true**.

> Setting the *subscript* property to **true** automatically sets the *base line offset* property to 0.3 (30 percent).

> Setting the *superscript* property to **true** automatically sets the *base line offset* property to -0.25 (-25 percent).

bold

> **True** if the character format is bold; **false** if it's not bold. Read/write.

east asian name

> Returns or sets the East Asian font name. Read/write.

embedable

> **True** if the specified font can be embedded in the presentation. This property should always return **false** because PowerPoint 2004 does not support embedded fonts. Read-only.

embedded

> **True** if the specified font is embedded in the presentation. This property should always return **false** because PowerPoint 2004 does not support embedded fonts. Read-only.

emboss

> **True** if the character format is embossed; **false** if it's not embossed. Read/write.

font color

> Returns an RGB color that represents the color for the specified font. Read/write.

font name

> Returns or sets the name of the font. Read\write.

font name other

> Returns or sets the roman font associated with text. Read/write.

font size

> Returns or sets the font size, in points. Read/write.

italic

> **True** if the character format is italic; **false** if it's not italic. Read/write.

shadow

> **True** if the specified text has a shadow; **false** if it doesn't have a shadow. This property does not affect soft shadows. Read/write.

subscript

> **True** if the specified text is subscript; **false** if it's not subscript. The default value is **false**. Read/write.

> Setting the *base line offset* property to a negative value automatically sets the *subscript* property to **true** and the *superscript* property to **false**.

> Setting the *base line offset* property to a positive value automatically sets the *subscript* property to **false** and the *superscript* property to **true**.

> Setting the *subscript* property to **true** automatically sets the *base line offset* property to -0.25 (-25 percent).

superscript

> **True** if the specified text is superscript; **false** if it's not superscript. The default value is **false**. Read/write.

> Setting the *base line offset* property to a negative value automatically sets the *subscript* property to **true** and the *superscript* property to **false**.

> Setting the *base line offset* property to a positive value automatically sets the *subscript* property to **false** and the *superscript* property to **true**.

> Setting the *superscript* property to **true** automatically sets the *base line offset* property to 0.3 (30 percent).

underline

> **True** if the specified text is underlined; **false** if it's not underlined. Read/write.

Class: header or footer

Represents a header, footer, date and time, slide number, or page number on a slide or master. All of the **header or footer** objects for a slide or master are contained in a **headers and footers** object.

To return the **header or footer** object, use one of the **headers and footers** object properties listed in the following table.

Use this property	To return
date and time	A **header or footer** object that represents the date and time on the slide.
footer	A **header or footer** object that represents the footer for the slide.
header	A **header or footer** object that represents the header for the slide. This works only for notes pages and handouts, not for slides.
slide number	A **header or footer** object that represent the slide number (on a slide) or page number (on a notes page or a handout).

Note   Header or footer objects aren't available for **slide** objects that represent notes pages. The **header or footer** object that represents a header is available only for a notes master or handout master.

You can set properties of **header or footer** objects for single slides. The following example sets the footer text for slide one in the active presentation.

```
set header footer text of footer of headers and footers of slide 1 of ¬
        active presentation to "Volcano Coffee"
```

You can also set properties of **header or footer** objects for the slide master, title master, notes master, or handout master to affect all slides, title slides, notes pages, or handouts and outlines at the same time. The following example sets the text for the footer in the slide master for the active presentation, sets the format for the date and time, and turns on the display of slide numbers. These settings will apply to all slides that are based on this master that display master graphics and that have not had their footer and date and time set individually.

```
set mySlidesHF to headers and footers of slide master of active presentation

set visible of footer of mySlidesHF to true

set header footer text of footer of mySlidesHF to "Regional Sales"

set visible of slide number of mySlidesHF to true

set visible of date and time of mySlidesHF to true

set use date format of date and time of mySlidesHF to true

set date format of date and time of mySlidesHF to Mdyy
```

Properties

date format

> Returns or sets the format for the automatically updated date and time. Applies only to **header or footer** objects that represent a date and time (returned from the **headers and footers** class by the *date and time* property). Read/write.

> Can be one of the following:

ddddMMMMddyyyy	**hmmssAMPM**
dMMMMyyyy	**Mdyy**
dMMMyy	**MMddyyHmm**
date time format unset	**MMddyyhmmAMPM**
Hmm	**MMMMdyyyy**
hmmAMPM	**MMMMyy**
Hmmss	**MMyy**

Make sure that the date and time are set to be updated automatically (not displayed as fixed text) by setting the *use date format* property to **true**.

header footer text

> Returns or sets the text contained in the **header or footer** object. Read/write.

use date format

> **True** if the date and time object contains automatically updated information; **false** if the date and time is a fixed string. Read/write.

> This property applies only to a **header or footer** object that represents a date and time (returned by the *date and time* property). Set the *use date format* property of a date and time **header or footer** object to **true** when you want to set or return the date and time format by using the *date format* property. Set the *use date format* property to **false** when you want to set or return the text string for the fixed date and time.

visible

> **True** if the specified header or footer, or the formatting applied to the specified header or footer, is visible. Read/write.

Class: headers and footers

Contains all the **header or footer** objects on the specified slide, notes page, handout, or master. Each **header or footer** object represents a header, footer, date and time, or slide number.

Note **Header or footer** objects aren't available for **slide** objects that represent notes pages. The **header or footer** object that represents a header is available only for a notes master or handout master.

To return the **headers and footers** object, use the *headers and footers* property of the **slide** class. To return an individual **header or footer** object, use the *date and time*, *footer*, *header*, or *slide number* property. The following example sets the footer text for slide one in the active presentation.

```
set header footer text of footer of headers and footers of slide 1 of ¬
        active presentation to "Volcano Coffee"
```

Properties

date and time

> Returns a **header or footer** object that represents the date and time item that appears in the lower-left corner of a slide or in the upper-right corner of a notes page, handout, or outline. Read-only.

display headers and footers on title slide

> **True** if the footer, date and time, and slide number appear on the title slide. **False** if this information appears on all slides except the title slide. Applies to slide masters. Read/write.

footer

> Returns a **header or footer** object that represents the footer that appears at the bottom of a slide or in the lower-left corner of a notes page, handout, or outline. Read-only.

header

> Returns a **header or footer** object that represents the header that appears at the top of a slide or in the upper-left corner of a notes page, handout, or outline. Read-only.

slide number

> Returns a **header or footer** object that represents the slide number in the lower-right corner of a slide, or the page number in the lower-right corner of a notes page or a page of a printed handout or outline. Read-only.

Class: hyperlink

Plural

hyperlinks

Represents a hyperlink associated with a shape. You can use a hyperlink to jump to an Internet or intranet site, to another file, or to a slide within the active presentation. The **hyperlinks** list contains all the hyperlinks on a slide or a master.

To return a hyperlink for a shape, use the *hyperlink* property of the **action setting** class. A shape can have two different hyperlinks assigned to it: one that's followed when the user clicks the shape during a slide show, and another that's followed when the user passes the mouse pointer over the shape during a slide show. For the hyperlink to be active during a slide show, the *action* property of the **action setting** class must be set to **action type hyperlink action**. The following example sets the mouse-click action for shape three on slide one in the active presentation to an Internet link.

```
set theAction to (get action setting for shape 3 of slide 1 ¬
        of active presentation event mouse activation mouse click)
set action of theAction to action type hyperlink action
set hyperlink address of hyperlink of theAction to "http://www.microsoft.com"
```

A slide can have more than one hyperlink. Use **hyperlink** *index*, where *index* is the hyperlink number, to return a single **hyperlink** object.

Properties

hyperlink address

> Returns or sets the address of the hyperlink. Read/write.

hyperlink sub address

> Returns or sets the location within a document — such as a bookmark in a Word document, a range in an Excel worksheet, or a slide in a PowerPoint presentation — associated with the specified hyperlink. Read/write.

hyperlink type

> Returns the type of hyperlink. Read-only.

> Can be one of the following:

>> - **hyperlink range**
>> - **hyperlink shape**
>> - **hyperlink inline shape**

Class: master

Elements

shape

hyperlink

Represents a slide master, title master, handout master, or notes master.

To return a **master** object, use the *slide master* property of the **slide** object, or use the *handout master*, *notes master*, *slide master*, or *title master* property of the **presentation** object. The following example sets the background fill for the slide master for the active presentation.

```
preset gradient (fill format of background of slide master ¬
        of active presentation) style horizontal gradient variant 1 ¬
        gradient type gradient brass
```

Properties

background

> Returns a **shape** object that represents the slide background. Read-only.
>
> If you want to use the *background* property to set the background for an individual slide without changing the slide master, the *follow master background* property for that slide must be set to **false**.

color scheme

> Returns or sets the **color scheme** object that represents the scheme colors for the specified slide master. Read/write.

headers and footers

> Returns a **headers and footers** object that represents the header, footer, date and time, and slide number associated with the slide master. Read-only.

height

> Returns or sets the height of the specified slide master. Read-only.

timeline

> Returns a **timeline** object representing the animation timeline for the slide master. Read-only.

width

> Returns or sets the width (in points) of the specified slide master. Read-only.

Class: motion effect

Represents a motion effect for an **animation behavior** object.

To return a **motion effect** object, use the *motion effect* property of the **animation behavior** object. The following example refers to the motion effect for a given animation behavior.

```
set MSQ to main sequence of timeline of slide 1 of active presentation
set theMotion to motion effect of animation behavior of effect 1 of MSQ
```

To construct a motion path, use the *by x*, *by y*, *from x*, *from y*, *to x*, and *to y* properties of the **motion effect** object. The following example adds a shape to the first slide and creates a motion path.

```
set theSlide to slide 1 of active presentation
set shpNew to make new shape at the beginning of theSlide with properties ¬
        {auto shape type:autoshape five point star, left position:0, top:0, ¬
        width:100, height:100}
set effNew to add effect (main sequence of timeline of theSlide) for shpNew ¬
        fx animation type custom trigger after previous
set aniMotion to add behavior effNew type animation type motion
set from x of motion effect of aniMotion to 0
set from y of motion effect of aniMotion to 0
set to x of motion effect of aniMotion to 100
set to y of motion effect of aniMotion to 100
```

Properties

by x

> Returns or sets a small real number that represents moving an object horizontally by a specified percentage of the screen width. For example, a value of 50 for a motion effect means to move the object half the screen width to the right. Read/write.

> Negative numbers move the object horizontally to the left. Floating point numbers (for example, 55.5) are allowed.

> To scale or move an object vertically, use the *by y* property.

> If both the *by x* and *by y* properties are set, then the object is scaled or moves both horizontally and vertically.

by y

> Returns or sets a small real number that represents moving an object vertically by a specified percentage of the screen width. Read/write.

> Negative numbers move the object horizontally to the left. Floating point numbers (for example, 55.5) are allowed.

> To scale or move an object horizontally, use the *by x* property.

> If both the *by x* and *by y* properties are set, then the object is scaled or moves both horizontally and vertically.

from x

>
> Returns or sets a small real number that represents the starting width or horizontal position of a **motion effect** object, specified as a percent of the screen width. Read/write.
>
> The default value of this property is **empty**, in which case the current position of the object is used.
>
> Use this property in conjunction with the *to x* property to resize or jump from one position to another.

from y

>
> Returns or sets a small real number that represents the starting height or vertical position of a **motion effect** object, specified as a percentage of the screen width. Read/write.
>
> The default value of this property is **empty**, in which case the current position of the object is used.
>
> Use this property in conjunction with the *to y* property to resize or jump from one position to another.

path

>
> Returns a string that represents the path followed by a **motion effect** object. The string returned is a specific path that the motion effect follows between from and to using the same syntax as the VML path description. Read-only.

to x

>
> Returns or sets a small real number that represents the ending width or horizontal position of a **motion effect** object, specified as a percent of the screen width. Read/write.
>
> The default value of this property is **empty**, in which case the current position of the object is used.
>
> Use this property in conjunction with the *from x* property to resize or jump from one position to another.

to y

>
> Returns or sets a small real number that represents the ending height or vertical position of a **motion effect** object, specified as a percentage of the screen width. Read/write.
>
> The default value of this property is **empty**, in which case the current position of the object is used.
>
> Use this property in conjunction with the *from y* property to resize or jump from one position to another.

Class: named slide show

Plural

named slide shows

Represents a custom slide show, which is a named subset of slides in a presentation. The **named slide shows** list contains all the named slide shows in the presentation.

Use **named slide show** *index*, where *index* is the custom slide show name or index number, to return a single **named slide show** object. The following example deletes the custom slide show named "Quick Show."

```
delete named slide show "Quick Show" of slide show settings ¬
        of active presentation
```

To return an array that contains the unique slide IDs for all the slides in the specified custom show, use the *slide IDs* property. The following example displays the slide IDs for the slides in the custom slide show named "Quick Show."

```
set idList to (get slide IDs of named slide show "Quick Show" of slide show ¬
        settings of active presentation)
repeat with i from 1 to (count of items of idList)
    display dialog item i of idList
end repeat
```

Properties

name

> Returns or sets the name of the specified named slide show. You cannot use this property to set the name for a custom slide show. To redefine a custom slide show under a new name, use the **make** command. Read-only.

number of slides

> Returns the number of slides in the specified named slide show. Read-only.

slide IDs

> Returns a list of slide IDs for the specified named slide show. Read-only.

Class: page setup

Contains information about the page setup for slides, notes pages, handouts, and outlines in a presentation.

To return the **page setup** object, use the *page setup* property of the **presentation** class. The following example sets all slides in the active presentation to be 11 inches wide and sets the slide numbering for the presentation to start at 17.

```
set slide width of page setup of active presentation to (11 * 72)
set first slide number of page setup of active presentation to 17
```

Properties

first slide number

Returns or sets the slide number for the first slide in the presentation. Read/write.

The slide number is the actual number that will appear in the lower-right corner of the slide when you display slide numbers. This number is determined by the number (order) of the slide within the presentation (the *slide index* property value) and the starting slide number for the presentation (the *first slide number* property value). The slide number will always be equal to the starting slide number + the slide index number - 1. The *slide number* property of the **slide** class returns the slide number.

notes orientation

Returns or sets the on-screen and printed orientation of notes pages, handouts, and outlines for the specified presentation. Read/write.

Can be one of the following:

- **horizontal orientation**
- **orientation unset**
- **vertical orientation**

slide orientation

Returns or sets the on-screen and printed orientation of slides in the specified presentation. Read/write.

Can be one of the following:

- **horizontal orientation**
- **orientation unset**
- **vertical orientation**

slide size

Returns or sets the slide size for the specified presentation. Read/write.

Can be one of the following:

- **slide size 35 MM**
- **slide size A4 paper**
- **slide size custom**
- **slide size letter paper**
- **slide size on screen**
- **slide size overhead**
- **slide size banner**

slide width

Returns or sets the slide width, in points. Read/write.

Class: pane

Plural

panes

An object representing one of the three panes in normal view or the single pane of any other view in the document window.

Use **pane** *index*, where *index* is the index number for a pane, to return a single **pane** object. The following table lists the names of the panes in normal view with their corresponding index numbers.

Pane	Index number
Outline	1
Slide	2
Notes	3

When you use a document window view other than normal view, use **pane** 1 to reference the single **pane** object.

The following example uses the *view type* property to return which pane is active.

```
get pane view type of active pane of active window
```

Remarks

Normal view is the only view with multiple panes. All other document window views have only a single pane, which is the document window.

Properties

active

> **True** if the specified pane is the active pane. Read-only.

pane view type

> Returns the type of the view contained in the specified pane. Read/write.

> Can be one of the following:

> - **normal view**
> - **handout master view**
> - **notes master view**
> - **page view**
> - **outline view**
> - **slide view**
> - **master view**
> - **slide sorter view**
> - **title master view**

Class: paragraph format

Represents the paragraph formatting of a text range.

To return the **paragraph format** object, use the *paragraph format* property of the **text range** or **text style level** class. The following example left aligns the paragraphs in shape two on slide one in the active presentation.

```
set alignment of paragraph format of text range of text frame of shape 2 of ¬
        slide 1 of active presentation to paragraph align left
```

Properties

alignment

> Returns or sets the alignment for each paragraph in the specified paragraph format. Read/write.

> Can be one of the following:

> - **paragraph align center**
> - **paragraph align distribute**
> - **paragraph align justify**
> - **paragraph align left**
> - **paragraph alignment unset**
> - **paragraph align right**

baseline alignment

> Returns or sets the base line alignment for the specified paragraph. Read/write.

> Can be one of the following:

> - **baseline align baseline**
> - **baseline center**
> - **baseline east asian50**
> - **baseline alignment unset**
> - **baseline align top**

bullet format

> Returns a **bullet format** object that represents bullet formatting for the specified paragraph format. Read-only.

east asian line break control

> **True** if you have an East Asian language setting specified **true** and the line break control option is set. Read/write.

line rule after

> **True** if line spacing after the last line in each paragraph is set to a specific number of lines, or **false** if line spacing is set to a specific number of points. Read/write.

line rule before

> **True** if line spacing before the first line in each paragraph is set to a specific number of lines, or **false** if line spacing is set to a specific number of points. Read/write.

line rule within

> **True** if line spacing between base lines is set to a specific number of lines, or **false** if line spacing is set to a specific number of points. Read/write.

space after

> Returns or sets the amount of space after the last line in each paragraph of the specified text, in points or lines. Read/write.

space within

> Returns or sets the amount of space between base lines in the specified text, in points or lines. Read/write.

text direction

> Returns or sets the text direction for the specified paragraph. Read/write.

> Can be one of the following:

> - **left to right**
> - **direction unset**

word wrap

> Used only with Kanji characters. Read/write.

Class: play settings

Contains information about how the specified media clip will be played during a slide show.

To return the **play settings** object, use the *animation play settings* property of the **animation settings** class or the *play settings information* property of the **effect information** class. The following sets the play settings for the animation associated with the specified shape.

```
set myAnim to animation settings of shape 2 of slide 1 of active presentation
set play on entry of play settings of myAnim to true
set pause animation of play settings of myAnim to false
set hide while not playing of play settings of myAnim to true
```

Properties

hide while not playing

> **True** if the specified media clip is hidden during a slide show except when it's playing. Read/write.

loop until stopped

> **True** if the specified movie or sound loops continuously until either the next movie or sound starts, the user clicks the slide, or a slide transition occurs. Read/write.

pause animation

> **True** if the slide show pauses until the specified media clip is finished playing. **False** if the slide show continues while the media clip plays in the background. Read/write.

> For the *pause animation* property setting to take effect, the *play on entry* property of the specified shape must be set to **true**.

play on entry

> **True** if the specified movie or sound is played automatically when it's animated. Read/write.

rewind move

> **True** if the first frame of the specified movie is automatically redisplayed as soon as the movie has finished playing. Read/write.

stop after slides

> Returns or sets the number of slides to be displayed before the media clip stops playing. Read/write.

> For the *stop after slides* property setting to take effect, the *pause animation* property of the specified slide must be set to **false**, and the *play on entry* property must be set to **true**.

> The media clip will stop playing when the specified number of slides have been displayed or when the clip comes to an end –– whichever comes first. A value of 0 (zero) specifies that the clip will stop playing after the current slide.

Class: presentation

Plural
presentations

Elements
slide
color scheme
font
document window
document property
custom document property

Represents a PowerPoint presentation. The **presentations** list contains all the **presentation** objects that represent open presentations in PowerPoint.

This section describes how to:

- Return a presentation that you specify by name or index number

- Return the presentation in the active window

- Return the presentation in any document window or slide show window you specify

Returning a presentation that you specify by name or index number

Use **presentation** *index*, where *index* is the presentation's name or index number, to return a single **presentation** object. The name of the presentation is the file name, with or without the file name extension, and without the path. The following example adds a slide to the beginning of Sample Presentation.

```
make new slide at the beginning of presentation "Sample Presentation" ¬
        with properties {slide index:1, layout:slide layout title slide}
```

Note that if multiple presentations with the same name are open, the first presentation in the collection with the specified name is returned.

Returning the presentation in the active window

To return the presentation in the active window, use the *active presentation* property. The following example saves the active presentation.

```
save active presentation
```

Note that if an embedded presentation is in-place active, the *active presentation* property returns the embedded presentation.

Returning the presentation in any document window or slide show window you specify

To return the presentation that's in the specified document window or slide show window, use the *presentation* property. The following example displays the name of the presentation in the first document window.

```
display dialog (get name of presentation of document window 1)
```

Properties

default shape

> Returns a **shape** object that represents the default shape for the presentation. Read-only.

east asian line break level

> Returns or sets the line break based upon Asian character level. Read/write.
>
> Can be one of the following:

>> • **east asian line break normal**
>>
>> • **east asian line break strict**
>>
>> • **east asian line break custom**

> This property is not used in the U.S. English version of PowerPoint.

full name

> Returns the name of the specified saved presentation, including the path, the current file system separator, and the file name extension. Read-only.
>
> This property is equivalent to the *path* property, followed by the current file system separator, followed by the *name* property.

handout master

Returns a **master** object that represents the handout master. Read-only.

has title master

True if the specified presentation has a title master. Read-only.

layout direction

Returns or sets the layout direction for the user interface. Read/write.

Can be one of the following:

- **left to right**

- **direction unset**

name

Returns the name of the specified object. The name of the presentation includes the file name extension (for file types that are registered) but doesn't include its path. You cannot use this property to set the name. Read-only.

no line break after

Returns or sets the characters that cannot end a line. Not used in the U.S. English version of PowerPoint. Read/write.

no line break before

Returns or sets the characters that cannot begin a line. Not used in the U.S. English version of PowerPoint. Read/write.

notes master

Returns a **master** object that represents the notes master. Read-only.

page setup

Returns a **page setup** object whose properties control slide setup attributes for the specified presentation. Read-only.

path

Returns the path to the specified **presentation** object. Read-only.

Note   The path doesn't include the final separator character or the name of the specified object. To return the file name without the path, use the *name* property. To return the file name and the path together, use the *full name* property.

print options

Returns a **print options** object that represents print options that are saved with the specified presentation. Read-only.

read only

> **True** if the specified presentation is read-only. Read-only.

save as movie settings

> Returns the **save as movie settings** object. When you derive a **save as movie settings** object using the *save as movie settings* property, any changes that you make to the properties of the **save as movie settings** object affect the specific presentation, not the default preferences for the application.

saved

> **True** if no changes have been made to a presentation since it was last saved. Read/write.

> If the *saved* property of a modified presentation is set to **true**, the user won't be prompted to save changes when closing the presentation, and all changes made to it since it was last saved will be lost.

slide master

> Returns a **master** object that represents the slide master. Read-only.

slide show settings

> Returns a **slide show settings** object that represents the slide show settings for the specified presentation. Read-only.

slide show window

> Returns a **slide show window** object that represents the slide show window in which the specified presentation is running. Read-only.

template name

> Returns the name of the design template associated with the specified presentation. Read-only.

> The returned string doesn't include the full path.

title master

> Returns a **master** object that represents the title master for the specified presentation. If the presentation doesn't have a title master, an error occurs. Read-only.

web options

> Returns the **web options** object, which contains presentation-level attributes used by PowerPoint when you save or publish a presentation as a Web page or open a Web page. Read-only.

Class: print options

Elements

print range

Contains print options for a presentation.

Note   Specifying the optional arguments **from**, **to**, **copies**, and **collate** for the **print out** command will set the corresponding properties of the **print options** object.

To return the **print options** object, use the *print options* property. The following example prints two uncollated color copies of all the slides (whether visible or hidden) in the active presentation. The example also scales each slide to fit the printed page and frames each slide with a thin border.

```
set number of copies of print options of active presentation to 2

set collate of print options of active presentation to false

set print color type of print options of active presentation to print color

set print hidden slides of print options of active presentation to true

set fit to page of print options of active presentation to true

set frame slides of print options of active presentation to true

set output type of print options of active presentation to print slides

print out active presentation
```

To specify whether to print the entire presentation or only a specified part of it, use the *range type* property. If you want to print only certain slides, set the *range type* property to **print range slide range**, and use the **print range** object to specify which pages to print. The following example prints slides 1, 4, 5, and 6 in the active presentation.

```
make new print range at the beginning of print options of active presentation ¬
        with properties {range start:1, range end:1}

make new print range at the beginning of print options of active presentation ¬
        with properties {range start:4, range end:6}

set range type of print options of active presentation ¬
        to print range slide range

print out active presentation
```

Properties

active printer

> Returns the name of the active printer. Read-only.

collate

> **True** if a complete copy of the specified presentation is printed before the first page of the next copy is printed. The default value is **true**. Read/write.

> Specifying a value for the **collate** argument of the **print out** command sets the value of this property.

fit to page

> **True** if the specified slides will be scaled to fill the page they're printed on, regardless of the values in the **Height** and **Width** boxes in the **Page Setup** dialog box (**File** menu). **False** if the slides will have the dimensions specified in the **Page Setup** dialog box, whether or not those dimensions match the page they're printed on. The default value is **false**. Read/write.

frame slides

> **True** if a thin frame is placed around the border of the printed slides. Applies to printed slides, handouts, and notes pages. The default value is **false**. Read/write.

number of copies

> Returns or sets the number of copies of a presentation to be printed. The default value is 1. Read/write.

> Specifying a value for the **copies** argument of the **print out** command sets the value of this property.

output type

> Returns or sets a value that indicates which component (slides, handouts, notes pages, or an outline) of the presentation is to be printed. Read/write.

> Can be one of the following:

> - **print notes pages**
> - **print outline**
> - **print four slide handouts**
> - **print nine slide handouts**
> - **print six slide handouts**
> - **print slides**
> - **print three slide handouts**
> - **print two slide handouts**

> The default value is **print slides**.

print color type

> Returns or sets the way the specified document will be printed: in black and white or in color. Read/write.

> Can be one of the following:

> - **print black and white**
> - **print color**

> The default value is set by the printer.

print fonts as graphics

> **True** if TrueType fonts are printed as graphics. Read/write.

range type

> Returns or sets the type of print range for the presentation. Read/write.

> Can be one of the following:

> - **print range all**
> - **print range current**
> - **print range selection**
> - **print range slide range**

slide show name

> Returns or sets the name of the custom slide show to print. Read/write.

Class: print range

Plural

print ranges

Represents a single range of consecutive slides or pages to be printed. The **print range** object is an element of the **print options** class. The **print ranges** list contains all the print ranges that have been defined for the specified presentation.

Use **print range** *index*, where *index* is the print range index number, to return a single **print range** object. The following example displays a message that indicates the starting and ending slide numbers for print range one in the active presentation.

```
set prtRanges to print ranges of print options of active presentation
if (count of prtRanges) > 0 then
    display dialog "Print range 1 starts on slide " & (range start of ¬
        item 1 of prtRanges) & " and ends on slide " & (range end of ¬
        item 1 of prtRanges)
end if
```

To create a **print range** object and add it to the **print ranges** list, use the **make** command. The following example defines three print ranges that represent slide 1, slides 3 through 5, and slides 8 and 9 in the active presentation and then prints the slides in these ranges.

```
make new print range at the beginning of print options of active presentation ¬
        with properties {range start:1, range end:1}
make new print range at the beginning of print options of active presentation ¬
        with properties {range start:3, range end:5}
make new print range at the beginning of print options of active presentation ¬
        with properties {range start:8, range end:9}
print out active presentation
```

Remarks

You can set print ranges in the **print ranges** list independent of the *range type* setting; these ranges are retained as long as the presentation they're contained in is loaded. The ranges in the **print ranges** list are applied when the *range type* property is set to **print range slide range**.

Properties

range end

> Returns the number of the last slide in the specified print range. Read-only.

range start

> Returns the number of the first slide in the range of slides to be printed. Read-only.

Class: property effect

Elements

animation point

Represents a property effect for an **animation behavior** object.

To return a **property effect** object, use the *property effect* property of the **animation behavior** object. The following example refers to the property effect for a specified animation behavior.

```
get property effect of animation behavior 1 of effect 1 of main sequence ¬
        of timeline of slide 1 of active presentation
```

Properties

ending

> Returns the ending value of the **property effect** object. Read-only.

property

> Returns or sets an animation property. Read/write.

> Can be one of the following:

x	**text font superscript**	**shape picture grayscale**
y	**text font underline**	**shape fill on**
opacity	**text font strikethrough**	**shape fill color**
colors	**text bullet character**	**shape fill opacity**
visibility	**text bullet fontName**	**shape fill back color**
text font bold	**text bullet number**	**shape line on**
text font color	**text bullet color**	**shape line color**
text font emboss	**text bullet relative size**	**shape shadow on**
text font italic	**text bullet style**	**shape shadow type**
text font name	**text bullet type**	**shape shadow color**
text font shadow	**shape picture contrast**	**shape shadow opacity**
text font size	**shape picture brightness**	**shape shadow offset X**
text font subscript	**shape picture gamma**	**shape shadow offset Y**

property effect set

> Returns an animation property. Read-only.

> Can be one of the following:

x	**text font superscript**	**shape picture grayscale**
y	**text font underline**	**shape fill on**
opacity	**text font strikethrough**	**shape fill color**
colors	**text bullet character**	**shape fill opacity**
visibility	**text bullet fontName**	**shape fill back color**
text font bold	**text bullet number**	**shape line on**
text font color	**text bullet color**	**shape line color**
text font emboss	**text bullet relative size**	**shape shadow on**
text font italic	**text bullet style**	**shape shadow type**
text font name	**text bullet type**	**shape shadow color**
text font shadow	**shape picture contrast**	**shape shadow opacity**
text font size	**shape picture brightness**	**shape shadow offset X**
text font subscript	**shape picture gamma**	**shape shadow offset Y**

starting

> Returns the starting value of the property effect object. Read-only.

> The default value is **empty**, in which case the current position of the object is used.

Class: rotating effect

Represents a rotation effect for an **animation behavior** object.

To return a **rotating effect** object, use the *rotating effect* property of the **animation behavior** object. The following example refers to the rotation effect for a given animation behavior.

```
get rotating effect of animation behavior 1 of effect 1 of main sequence ¬

        of timeline of slide 1 of active presentation
```

Properties

rotating

Class: ruler

Elements
tab stop

ruler level

Represents the ruler for the text in the specified shape or for all text in the specified text style. Contains tab stops and the indentation settings for text outline levels.

To return the **ruler** object that represents the ruler for the text in the specified shape, use the *ruler* property of the **text frame** class. The following example sets a left-aligned tab stop at 2 inches (144 points) and sets a hanging indent for the text in object two on slide one in the active presentation.

```
set theRuler to ruler of text frame of shape 2 of slide 1 of ¬
        active presentation
make new tab stop at theRuler with properties {tab stop type:tabstop left, ¬
        tab position:144}
set first margin of ruler level 1 of theRuler to 0
set left margin of ruler level 1 of theRuler to 36
```

To return the **ruler** object that represents the ruler for one of the four defined text styles (title text, body text, notes text, or default text), use the *ruler* property of the **text style** class. The following example sets the first-line indent and hanging indent for outline level one in body text on the slide master for the active presentation.

```
set tsBody to get text style from slide master of active presentation ¬
        at text style body
set rlOne to ruler level 1 of ruler of tsBody
set first margin of rlOne to 9
set left margin of rlOne to 54
```

Class: ruler level

Plural

ruler levels

Contains first-line indent and hanging indent information for an outline level. The **ruler levels** list contains a **ruler level** object for each of the five available outline levels.

Use **ruler level** *index*, where *index* is the outline level, to return a single **ruler level** object. The following example sets the first-line indent and hanging indent for outline level one in body text on the slide master for the active presentation.

```
set tsBody to get text style from slide master of active presentation ¬
        at text style body
set first margin of ruler level 1 of ruler of tsBody to 9
set left margin of ruler level 1 of ruler of tsBody to 54
```

The following example sets the first-line indent and hanging indent for outline level one in shape two on slide one in the active presentation.

```
set rlOne to ruler level 1 of ruler of text frame of shape 2 of ¬
        slide master of active presentation
set first margin of rlOne to 9
set left margin of rlOne to 54
```

Properties

first margin

Returns or sets the first-line indent (in points) for the specified outline level. Read/write.

If a paragraph begins with a bullet, the bullet position is determined by the *first margin* property, and the position of the first text character in the paragraph is determined by the *left margin* property.

Note The **ruler levels** list contains five **ruler level** objects, each of which corresponds to one of the possible outline levels. The *first margin* property value for the **ruler level** object that corresponds to the first outline level has a valid range of (-9.0 to 4095.875). The valid range for the *first margin* property values for the **ruler level** objects that correspond to the second through the fifth outline levels are determined as follows:

- The maximum value is always 4095.875.

- The minimum value is the maximum assigned value between the *first margin* property and *left margin* property of the previous level plus 9.

left margin

Returns or sets the left indent (in points) for the specified outline level. Read/write.

If a paragraph begins with a bullet, the bullet position is determined by the *first margin* property, and the position of the first text character in the paragraph is determined by the *left margin* property.

Note The **ruler levels** list contains five **ruler level** objects, each of which corresponds to one of the possible outline levels. The *left margin* property value for the **ruler level** object that corresponds to the first outline level has a valid range of (-9.0 to 4095.875). The valid range for the *left margin* property values for the **ruler level** objects that correspond to the second through the fifth outline levels are determined as follows:

- The maximum value is always 4095.875.

- The minimum value is the maximum assigned value between the *first margin* property and *left margin* property of the previous level plus 9.

Class: save as movie settings

Contains the settings for saving presentations as movies.

To return the **save as movie settings** object from a **presentation** object, use the *save as movie settings* property of the **presentation** class. Any changes that you make to the properties of the **save as movie settings** object affect the specific presentation, not the default preferences for the application.

To return the **save as movie settings** object from the **application** object, use the *save as movie settings object* property of the **application** class. Any changes that you make to the properties of the **save as movie settings** object affect the default preferences for the application, not the properties of a specific presentation.

Properties

animation enabled

> **True** if animation is enabled. Read/write.

auto loop enabled

> **True** if the movie restarts after it has completed. Read/write.

background sound track file

> Specifies the location of the sound file for the presentation. Read/write.

background track segment end

> Specifies the end location, in seconds, for the sound relative to the beginning of the sound file. Read/write.

background track segment start

> Specifies the start location, in seconds, for the sound relative to the beginning of the sound file. Read/write.T

background track start

> Specifies the start location, in seconds, for the sound track relative to the beginning of the movie. Read/write.

create movie preview

> **True** to create a preview of a movie. Read/write.

force all inline

> Sets the size of the movie to the size of the slide. Read/write.

include narration and sounds

> **True** to include narration and sounds with a movie. Read/write.

include roundtrip data

> **True** to include data that would allow subsequent editing of a movie in PowerPoint. Read/write.

interactivity enabled

> **True** to allow interactivity, such as hyperlinks, with the saved movie. Read/write

movie actors

> Sets a text description that is included with a movie. Read/write.

movie author

> Sets a text description that is included with a movie. Read/write.

movie copyright

> Sets a text description that is included with a movie. Read/write.

movie frame height

> Sets the height of a movie's frame. Read/write.

movie frame width

> Sets the width of a movie's frame. Read/write.

movie producer

> Sets a text description that is included with a movie. Read/write.

optimization

> Specifies the type of optimization to use when creating a movie. Read/write.

> Can be one of the following:

- **movie optimization normal**
- **movie optimization size**
- **movie optimization speed**
- **movie optimization quality**

show movie controller

> **True** if the movie controller is made visible when a movie is being created. Read/write.

transition description

> Sets the description for the transition type used in a movie. The description should correspond to one of three transition types: Follow slideshow, No transitions, or MRU QT transition. Read/write.

> The possible transition types correspond to the options available in the **Movie Options** dialog box.

> If the description is set to MRU QT transition, then the QuickTime transition that was last selected for a movie is used.

use single transition

> **True** to use a single QuickTime transition throughout a movie. Read/write.

Class: scale effect

Represents a scaling effect for an **animation behavior** object.

Use the *scale effect* property of the **animation behavior** object to return a **scale effect** object. The following example refers to the scale effect for a given animation behavior.

```
get scale effect of animation behavior 1 of main sequence of timeline of ¬
        slide 1 of active presentation
```

Use the *by x*, *by y*, *from x*, *from y*, *to x*, and *to y* properties of the **scale effect** object to manipulate an object's scale. This example scales the first shape on the first slide, starting at zero then increasing in size until it reaches 100 percent of its original size. This example assumes that there is a shape on the first slide.

```
set shpFirst to shape 1 of slide 1 of active presentation

set effNew to add effect (main sequence of timeline of slide 1 of ¬
        active presentation) for shpFirst fx animation type custom

set aniScale to add behavior effNew type animation type scale

set from x of scale effect of aniScale to 0

set from y of scale effect of aniScale to 0

set to x of scale effect of aniScale to 100

set to y of scale effect of aniScale to 100
```

Properties

by x

> Returns or sets a small real number that represents scaling an object horizontally by a specified percentage of the screen width. For example, a value of 50 for a motion effect means to scale the object half the screen width to the right. Read/write.

> Negative numbers scale the object horizontally to the left. Floating point numbers (for example, 55.5) are allowed.

> To scale an object vertically, use the *by y* property.

> If both the *by x* and *by y* properties are set, then the object is scaled both horizontally and vertically.

by y

> Returns or sets a small real number that represents moving an object vertically by a specified percentage of the screen width. Read/write.

> Negative numbers scale the object vertically downward. Floating point numbers (for example, 55.5) are allowed.

> To scale an object horizontally, use the *by x* property.

> If both the *by x* and *by y* properties are set, then the object is scaled both horizontally and vertically.

from x

> Returns or sets a small real number that represents the starting width or horizontal position of a **scale effect** object, specified as a percent of the screen width. Read/write.

> The default value of this property is **empty**, in which case the current position of the object is used.

> Use this property in conjunction with the *to x* property to resize or jump from one position to another.

from y

> Returns or sets a small real number that represents the starting height or vertical position of a **scale effect** object, specified as a percentage of the screen height. Read/write.

> The default value of this property is **empty**, in which case the current position of the object is used.

> Use this property in conjunction with the *to y* property to resize or jump from one position to another.

to x

> Returns or sets a small real number that represents the ending width or horizontal position of a **scale effect** object, specified as a percent of the screen width. Read/write.

> The default value of this property is **empty**, in which case the current position of the object is used.

> Use this property in conjunction with the *from x* property to resize or jump from one position to another.

to y

> Returns or sets a small real number that represents the ending height or vertical position of a **scale effect** object, specified as a percentage of the screen height. Read/write.

> The default value of this property is **empty**, in which case the current position of the object is used.

> Use this property in conjunction with the *from y* property to resize or jump from one position to another.

Class: sequence

Plural

sequences

Elements

effect

Represents a slide's interactive animation sequences. The **sequence** class is an element of the **timeline** class.

To return a **sequence** object, use the *main sequence* property of the **timeline** object.

To add a new **sequence** object, use the **add effect** command. This example adds a shape and an animation sequence to the first shape on the first slide in the active presentation.

```
set shpFirst to shape 1 of slide 1 of active presentation
set effNew to add effect (main sequence of timeline of slide 1 of ¬
        active presentation) for shpFirst fx animation type blinds
```

Class: set effect

Represents a set effect for an animation behavior. You can use the **set effect** object to set the value of a property.

Use the *set effect* property of the **animation behavior** object to return a **set effect** object. The following example adds a shape to the first slide of the active presentation and sets a set effect animation behavior.

```
set sldFirst to slide 1 of active presentation
set shpHeart to make new shape at the beginning of sldFirst with properties ¬
        {auto shape type:autoshape heart, left position:100, top:100, ¬
        width:100, height:100}
set effNew to add effect (main sequence of timeline of sldFirst) for shpHeart ¬
        fx animation type change fill color trigger after previous
set bhvEffect to add behavior effNew type animation type set
set property set effect of set effect of bhvEffect to shape fill color
set ending of set effect of bhvEffect to ({0, 255, 255} as RGB color)
```

Properties

ending

      Sets or returns the value or ending value of the **set effect** object. Read/write.

property set effect

      Returns an animation property. Read-only.

      Can be one of the following:

x	**text font superscript**	**shape picture grayscale**
y	**text font underline**	**shape fill on**
opacity	**text font strikethrough**	**shape fill color**
colors	**text bullet character**	**shape fill opacity**
visibility	**text bullet fontName**	**shape fill back color**
text font bold	**text bullet number**	**shape line on**
text font color	**text bullet color**	**shape line color**
text font emboss	**text bullet relative size**	**shape shadow on**
text font italic	**text bullet style**	**shape shadow type**
text font name	**text bullet type**	**shape shadow color**
text font shadow	**shape picture contrast**	**shape shadow opacity**
text font size	**shape picture brightness**	**shape shadow offset X**
text font subscript	**shape picture gamma**	**shape shadow offset Y**

Class: slide

Plural
slides

Elements
shape

hyperlink

Represents a slide. The **slides** list contains all the **slide** objects in a presentation.

This section describes how to:

- Return a slide that you specify by name, index number, or slide ID number.

- Return the slide that's currently displayed in any document window or slide show window you specify.

- Create a new slide.

Returning a slide that you specify by name, index number, or slide ID number

Use **slide** *index*, where *index* is the slide name or index number to return a single **slide** object. The following example sets the layout for slide one in the active presentation.

```
set layout of slide 1 of active presentation to slide layout title slide
```

Returning the slide that's currently displayed in any document window or slide show window you specify

Use the *slide* property to return the slide that's currently displayed in the specified document window or slide show window view. The following example copies the slide that's currently displayed in document window two to the Clipboard.

```
copy object (slide of view of window 2)
```

Creating a new slide

Use the **make** command to create a new slide and add it to the presentation. The following example adds a title slide to the beginning of the active presentation.

```
make new slide at the beginning of the active presentation with properties ¬
        {layout:slide layout title slide}
```

Properties

background

> Returns a **shape** object that represents the slide background. Read-only.
>
> If you want to use the *background* property to set the background for an individual slide without changing the slide master, the *follow master background* property for that slide must be set to **false**.

color scheme

> Returns or sets the **color scheme** object that represents the scheme colors for the specified slide, slide range, or slide master. Read/write.

display master shapes

> **True** if the specified slide displays the background objects on the slide master. These background objects can include text, drawings, and clip art you add to the slide master. Headers and footers aren't included. Read/write.
>
> When you create a new slide, the default value for this property is **true**. If you copy a slide from another presentation, it retains the setting it had in the original presentation. That is, if the slide omitted slide master background objects in the original presentation, it will omit them in the new presentation as well.
>
> Note that the look of the slide's background is determined by the color scheme and background as well as by the background objects. If setting the *display master shapes* property alone doesn't give you the results you want, try setting the *follow master background* and *color scheme* properties as well.

follow master background

> **True** if the specified slide or range of slides follows the slide master background. **False** if the specified slide or range of slides has a custom background. Read/write.
>
> When you create a new slide, the default value for this property is **true**. If you copy a slide from another presentation, it retains the setting it had in the original presentation. That is, if the slide followed the slide master background in the original presentation, it will automatically follow the slide master background in the new presentation; or, if the slide had a custom background, it will retain that custom background.
>
> Note that the look of the slide's background is determined by the color scheme and background objects as well as by the background itself. If setting the *follow master background* property alone doesn't give you the results you want, try setting the *color scheme* and *display master shapes* properties as well.

headers and footers

> Returns a **headers and footers** object that represents the header, footer, date and time, and slide number associated with the slide. Read-only.

layout

Returns or sets the slide layout. Read/write.

Can be one of the following:

slide layout title slide	**slide layout large object**
slide layout text slide	**slide layout object**
slide layout two column text	**slide layout media clip**
slide layout table	**slide layout media clip and text**
slide layout text and chart	**slide layout object over text**
slide layout chart and text	**slide layout text over object**
slide layout orgchart	**slide layout text and two objects**
slide layout chart	**slide layout two objects and text**
slide layout text and clipart	**slide layout two objects over text**
slide layout clipart and text	**slide layout four objects**
slide layout title only	**slide layout vertical text**
slide layout blank	**slide layout clipart and vertical text**
slide layout text and object	**slide layout vertical title and text over chart**
slide layout object and text	

notes page

Returns a **slide** object that represents the notes pages for the specified slide or range of slides. Read-only.

The *notes page* property returns the notes page for a single slide and allows you to make changes only to those notes pages. To make changes that affect all notes pages, use the *notes master* property to return the **slide** object that represents the notes master.

print steps

Returns the number of slides you'd need to print to simulate the builds on the specified slide. Read-only.

slide ID

Returns a unique ID number for the specified slide. Read-only.

Unlike the *slide index* property, the *slide ID* property of a **slide** object won't change when you add slides to the presentation or rearrange the slides in the presentation.

slide index

Returns the index number of the specified slide within the **slides** list. Read-only.

Unlike the *slide ID* property, the *slide index* property of a **slide** object can change when you add slides to the presentation or rearrange the slides in the presentation.

slide master

Returns a **master** object that represents the slide master. Read-only.

slide number

> Returns the slide number. Read-only.

> The *slide number* property of a **slide** object is the actual number that appears in the lower-right corner of the slide when you display slide numbers. This number is determined by the number of the slide within the presentation (the *slide index* property value) and the starting slide number for the presentation (the *first slide number* property value). The slide number is always equal to the the starting slide number + the slide index number - 1.

slide show transition

> Returns a **slide show transition** object that represents the special effects for the specified slide transition. Read-only.

timeline

> Returns a **timeline** object representing the animation timeline for the slide. Read-only.

Class: slide show settings

Elements
named slide show

Represents the slide show setup for a presentation.

Use the *slide show settings* property of the **presentation** class to return the **slide show settings** object. The first section in the following example sets all the slides in the active presentation to advance automatically after five seconds. The second section sets the slide show to start on slide two, end on slide four, advance slides by using the timings set in the first section, and run in a continuous loop until the user presses ESC. Finally, the example runs the slide show.

```
repeat with s in (get slides of active presentation)
    set advance on time of slide show transition of s to true
    set advance time of slide show transition of s to 5
end repeat


set theSlideShowSet to slide show settings of active presentation
set starting slide of theSlideShowSet to 2
set ending slide of theSlideShowSet to 4
set advance mode of theSlideShowSet to slide show advance use slide timings
set loop until stopped of theSlideShowSet to true
run slide show theSlideShowSet
```

Properties

advance mode

Returns or sets a value that indicates how the slide show advances. Read/write.

Can be one of the following:

- **slide show advance manual advance**

- **slide show advance use slide timings**

ending slide

Returns or sets the last slide to be displayed in the specified slide show. Read/write.

loop until stopped

True if the specified slide show loops continuously until the user presses ESC. Read/write.

pointer color

Returns or sets the pointer color for the specified presentation as an RGB color. This color is saved with the presentation and is the default pen color each time you show the presentation. Read/write.

range type

Returns or sets the type of slide show to run. Read/write.

Can be one of the following:

- **slide show range show all**

- **slide show range named slideshow**

- **slide show range**

show type

Returns or sets the show type for the specified slide show. Read/write.

Can be one of the following:

- **slide show type speaker**

- **slide show type window**

show with animation

True if the specified slide show displays shapes with assigned animation settings. Read/write.

show with narration

True if the specified slide show is shown with narration. Read/write.

slide show name

Returns the name of the custom slide show that's currently running in the specified slide show view. Read-only.

starting slide

Returns or sets the first slide to be displayed in the specified slide show. Read/write.

Class: slide show transition

Contains information about how the specified slide advances during a slide show.

Use the *slide show transition* property of the **slide** class to return the **slide show transition** object. The following example specifies a Fast Strips Down-Left transition accompanied by the Bass.au sound for slide one in the active presentation and specifies that the slide advance automatically five seconds after the previous animation or slide transition.

```
set slideTransit to slide show transition of slide 1 of active presentation
set entry effect of slideTransit to entry effect strips left down
import sound file sound effect of slideTransit ¬
        sound file name "Macintosh HD:Users:Shared:Bass.au"
set advance on time of slideTransit to true
set advance time of slideTransit to 5
set advance mode of slide show settings of active presentation to ¬
        slide show advance use slide timings
```

Properties

advance on click

> **True** if the specified slide advances when it's clicked during a slide show. Read/write.

> If you set both the *advance on click* and the *advance on time* properties to **true**, the slide will advance either when it's clicked or when the specified amount of time has elapsed — whichever comes first.

advance on time

> **True** if the specified slide advances automatically after a specified amount of time has elapsed. To specify the number of seconds after which the slide will automatically advance, use the *advance time* property. Read/write.

> To put the slide interval settings into effect for the entire slide show, set the *advance mode* property of the **slide show settings** object to **slide show advance use slide timings**.

advance time

> Returns or sets the amount of time after which the specified slide transition will occur. Read/write.

> The specified slide transition won't advance automatically unless the *advance mode* property of the slide show settings is set to **slide show advance use slide timings**.

entry effect

Returns or sets the special effect applied to the specified slide transition. Read/write.

Can be one of the following:

entry effect appear
entry effect horizontal
entry effect blinds vertical
entry effect box in
entry effect box out
entry effect checkerboard across
entry effect checkerboard down
entry effect circle
entry effect comb horizontal
entry effect comb vertical
entry effect cover down
entry effect cover left
entry effect cover left down
entry effect cover left up
entry effect cover right
entry effect cover right down
entry effect cover right up
entry effect cover up
entry effect crawl from down
entry effect crawl from left
entry effect crawl from right
entry effect crawl from up
entry effect cube down
entry effect cube left
entry effect cube right
entry effect cube up
entry effect cut
entry effect cut through black
entry effect diamond
entry effect dissolve
entry effect fade
entry effect flash once fast
entry effect flash once medium
entry effect flash once slow
entry effect flip down
entry effect flip left
entry effect flip right
entry effect flip up
entry effect fly from bottom
entry effect fly from bottom left
entry effect fly from bottom right
entry effect fly from left
entry effect fly from right
entry effect fly from top

entry effect fly from top left
entry effect fly from top right
entry effect unset
entry effect none
entry effect peek from down
entry effect peek from left
entry effect peek from right
entry effect peek from up
entry effect plus
entry effect random
entry effect random bars horizontal
entry effect random bars vertical
entry effect spinner
entry effect split horizontal in
entry effect split horizontal out
entry effect split vertical in
entry effect split vertical out
entry effect strips down left
entry effect strips down right
entry effect strips left down
entry effect strips left up
entry effect strips right down
entry effect strips right up
entry effect strips up left
entry effect strips up right
entry effect uncover down
entry effect uncover left
entry effect uncover left down
entry effect uncover left up
entry effect uncover right
entry effect uncover right down
entry effect uncover right up
entry effect uncover up
entry effect wedge
entry effect wheel1 spoke
entry effect wheel2 spokes
entry effect wheel3 spokes
entry effect wheel4 spokes
entry effect wheel8 spokes
entry effect wipe down
entry effect wipe left
entry effect wipe right
entry effect wipe up

Remarks

If the *text level effect* property for the specified animation is set to **animate level none** (the default value) or the *animate* property is set to **false**, you won't see the special effect you've applied with the *entry effect* property.

hidden

True if the specified slide is hidden during a slide show. Read/write.

loop sound until next

True if the sound that's been set for the specified slide transition loops until the next sound starts. Read/write.

sound effect transition

Returns a **sound effect** object that represents the sound to be played during the transition to the specified slide. Read-only.

Class: slide show view

Represents the view in a slide show window.

Use the *slideshow view* property of the **slide show window** object to return the **slide show view** object.

Use the **run slide show** command to create a **slide show window** object, and then use the *view* property to return the **slide show view** object the window contains. The following example runs a slide show of the active presentation and changes the pointer to a pen.

```
set theWindow to run slide show slide show settings of active presentation
set theView to slide show view of theWindow
set pointer type of theView to slide show pointer pen
```

Properties

accelerations enabled

True if shortcut keys are enabled during a slide show. The default value is **true**. If shortcut keys are disabled during a slide show, you can neither use keys to navigate in the slide show nor press ESC to exit the slide show. Read/write.

current show position

Returns the position of the current slide within the slide show that is showing in the specified view. Read-only.

If the specified view contains a custom show, the *current show position* property returns the position of the current slide within the custom show, not the position of the current slide within the entire presentation.

is named show

True if a custom (named) slide show is displayed in the specified slide show view. Read-only.

last slide viewed

Returns a **slide** object that represents the slide viewed immediately before the current slide in the specified slide show view. Read-only.

pointer color

Returns an RGB color that represents the pointer color for the specified presentation during one slide show. As soon as the slide show is finished, the color reverts to the default color for the presentation. Read-only.

To change the pointer to a pen, set the *pointer type* property to **slide show pointer pen**.

pointer type

Returns or sets the type of pointer used in the slide show. Read/write.

Can be one of the following:

- **slide show pointer always hidden**
- **slide show pointer arrow**
- **slide show pointer none**
- **slide show pointer pen**

presentation elapsed time

Returns the number of seconds that have elapsed since the beginning of the specified slide show. Read-only.

slide show name

Returns the name of the custom slide show that's currently running in the specified slide show view. Read-only.

slide state

Returns or sets the state of the slide show. Read/write.

Can be one of the following:

- **slide show state black screen**
- **slide show state paused**
- **slide show state running**
- **slide show state white screen**

zoom

Returns the zoom setting of the specified slide show window view as a percentage of normal size. Can be a value from 10 to 400. Read-only.

Class: slide show window

Plural

slide show windows

Represents a window in which a slide show runs. The **slide show windows** list contains all the open slide show windows.

Use **slide show window** *index*, where *index* is the slide show window index number, to return a single **slide show window** object.

Use the **run slide show** command to create a new slide show window and return a reference to this slide show window. The following example runs a slide show of the active presentation and reduces the height of the slide show window just enough so that you can see the taskbar (for screens with a resolution of 800 by 600).

```
set show type of slide show settings of active presentation to ¬
        slide show type speaker
set theSSW to run slide show slide show settings of active presentation
set height of theSSW to 300
set width of theSSW to 400
```

Use the *presentation* property to return the presentation that's currently running in the specified slide show window. The following example displays the name of the presentation that's currently running in slide show window one.

```
display dialog (get name of presentation of slide show window 1)
```

Properties

active

> **True** if the specified slide show window is the active window. Read-only.

bounds

> Returns a bounding rectangle that defines the slide show window. Read-only.

height

> Returns or sets the height (in points) of the **slide show window** object. Read/write.

is full screen

> **True** if the specified slide show window occupies the full screen. Read-only.

left position

> Returns or sets the distance (in points) from the left edge of the slide show window to the left edge of the desktop. Setting this property to a very large positive or negative value may position the window completely off the desktop. Read/write.

presentation

> Returns a **presentation** object that represents the presentation in which the specified document window or slide show window was created. Read-only.

slideshow view

> Returns a **slide show view** object. Read-only.

top

> Returns or sets the distance (in points) from the top edge of the application window or slide show window to the top edge of the desktop. Setting this property to a very large positive or negative value may position the window completely off the desktop. Read/write.

width

> Returns or sets the width (in points) of the specified slide show window object. Read/write.

Class: sound effect

Represents the sound effect that accompanies an animation or slide transition in a slide show.

To return the **sound effect** object that represents the sound effect that accompanies an animation, use the *animation sound effect* property of the **animation settings** object. The following example specifies that the animation of the title on slide one in the active presentation be accompanied by the sound in the Bass.au file.

```
set theAnim to animation settings of shape 1 of slide 1 of active presentation
set text level effect of theAnim to animate level all levels
import sound file animation sound effect of theAnim ¬
        sound file name "Macintosh HD:Users:Shared:Bass.au"
```

To return the **sound effect** object that represents the sound effect that accompanies a slide transition, use the *sound effect transition* property of the **slide show transition** object.

The following example specifies that the transition to slide one in the active presentation be accompanied by the sound in the Bass.wav file.

```
set theSE to sound effect transition of slide show transition of slide 1 of ¬
        active presentation
import sound file theSE sound file name "Macintosh HD:Users:Shared:Bass.wav"
```

Properties

name

> Returns or sets the name of the specified sound effect. The set of valid names for a presentation appears on the the **Sound** pop-up menu in the **Slide Transition** dialog box (**Slide Show** menu). Read/write.

sound type

> Returns or sets the sound type of the **sound effect** object.
>
> Can be one of the following:
>
> - **sound effect unset**
> - **sound effect none**
> - **sound effect stop previous**
> - **sound effect file**

Class: tab stop

Plural

tab stops

Represents a single tab stop. The **tab stops** list represents all the tab stops on one ruler.

The **tab stop** class is an element of the **ruler** class. Use **tab stop** *index*, where *index* is the tab stop index number, to return a single **tab stop** object. The following example clears tab stop one for the text in shape two on slide one in the active presentation.

```
delete tab stop 1 of ruler of text frame of shape 2 of slide 2 ¬
        of active presentation
```

Properties

tab position

Returns or sets the position (in points) of the specified tab stop. Read/write.

tab stop type

Returns or sets the tab stop type of object. Read/write.

Can be one of the following:

- **tabstop unset**

- **tabstop left**

- **tabstop center**

- **tabstop right**

Class: text style

Elements

text style level

Represents one of three text styles: title text, body text, or default text. Each text style contains a **text frame** object that describes how text is placed within the text bounding box, a **ruler** object that contains tab stops and outline indent formatting information, and a **text style level** object that contains outline text formatting information.

Use the **get text style from** command to return a single **text style** object from a slide master. The following example sets the font name and font size for level-one body text on all the slides in the active presentation.

```
set tsBody to get text style from slide master of active presentation ¬
        at text style body
set font name of font of text style level 1 of tsBody to "Arial"
set font size of font of text style level 1 of tsBody to 36
```

Properties

ruler

> Returns a **ruler** object that represents the ruler for the specified text. Read-only.

text frame

> Returns a **text frame** object that contains the alignment and anchoring properties for the specified shape or master text style. Read-only.

Class: text style level

Plural

text style levels

Contains character and paragraph formatting information for an outline level. The **text style levels** list contains one **text style level** object for each of the five outline levels.

Use **text style level** *index*, where *index* is a number from 1 through 5 that corresponds to the outline level, to return a single **text style level** object. The following example sets the font name and font size, the space after paragraphs, and the paragraph alignment for level-one body text on all the slides in the active presentation.

```
set tsBody to get text style from slide master of active presentation ¬
        at text style body
set tlOne to text style level 1 of tsBody
set font name of font of tlOne to "Arial"
set font size of font of tlOne to 36
set line rule before of paragraph format of tlOne to false
set space after of paragraph format of tlOne to 14
set alignment of paragraph format of tlOne to paragraph align justify
```

Properties

font

> Returns a **font** object that represents character formatting. Read-only.

paragraph format

> Returns a **paragraph format** object that represents paragraph formatting for the specified text. Read-only.

Class: timeline

Elements

sequence

Stores animation information for a **master** or **slide**.

Use the *timeline* property of the **master** or **slide** object to return a **timeline** object.

The **timeline** object's *main sequence* property gains access to the main animation sequence. The following example returns the main sequence of the timeline of the first slide in the presentation:

```
set MSQ to main sequence of timeline of slide 1 of active presentation
```

Properties

main sequence

> Returns a **sequence** object that represents the list of **effect** objects in the main animation sequence of a slide.

> The default value of the *main sequence* property is an empty **sequences** list. Any attempt to return a value from this property without adding one or more **effect** objects to the main animation sequence will result in a run-time error.

Class: timing

Represents timing properties for an animation effect.

To return a **timing** object, use the *timing* property of the **animation behavior** or **effect** object. The following example sets timing duration information for the main animation.

```
set MSQ to main sequence of timeline of slide 1 of active presentation
set theEffect to effect 1 of MSQ
set duration of timing of theEffect to 5
```

Use the following read/write properties of the **timing** object to manipulate animation timing effects.

Properties

acceleration

> Returns or sets a small real number that represents the percent of the duration over which a timing acceleration should take place. For example, a value of 0.9 means that an acceleration should start slower than the default speed for 90% of the total animation time, with the last 10% of the animation at the default speed. Read/write.

> To slow down an animation at the end, use the *deceleration* property.

autoreverse

> Returns or sets whether an effect should play forward and then reverse, thereby doubling the duration. Read/write.

deceleration

Returns or sets a small real number that represents the percent of the duration over which a timing deceleration should take place. For example, a value of 0.9 means that a deceleration should start at the default speed, and then start to slow down after the first ten percent of the animation. Read/write.

duration

Returns or sets a small real number that represents the length of an animation in seconds. Read/write.

repeat count

Returns or sets an integer that represents the number of times to repeat an animation. Read/write.

repeat duration

Returns or sets a small real number that represents, in seconds, how long repeated animations should last. Read/write.

restart

Returns or sets a constant that represents whether the animation effect restarts after the effect has started once. Read/write.

Can be one of the following:

- **restart always**
- **restart when off**
- **never restart**

rewind

Returns or sets whether an object returns to its beginning position after an animation has ended. Read/write.

smooth end

Returns or sets whether an animation should decelerate as it ends. Read/write.

smooth start

Returns or sets whether an animation should accelerate when it starts. Read/write.

speed

Returns or sets a small real number that represents the speed, in seconds, of the specified animation. Read/write.

Class: view

Represents the current editing view in the specified document window.

Use the *view* property of the **document window** object to return the **view** object. The following example sets the size of window one and then sets the zoom to fit the new window size.

```
set height of document window 1 to 200
```

```
set width of document window 1 to 250
```

```
set zoom to fit of view of document window 1 to true
```

Note   The **view** object can represent any of the document window views: slide view, outline view, slide sorter view, notes page view, slide master view, handout master view, or notes master view. Some properties and methods of the **view** object work only in certain views. If you try to use a property or command that's inappropriate for a **view** object, an error occurs.

Properties

display slide miniature

> **True** if the slide miniature window is displayed. Read/write.

slide

> Returns or sets a **slide** object that represents the slide that's currently displayed in the specified document window view. Read/write.

view type

> Returns or sets the type of the view contained in the specified view. Read/write.

> Can be one of the following:

> - **page view**
> - **handout master view**
> - **notes master view**
> - **normal view**
> - **outline view**
> - **slide view**
> - **master view**
> - **slide sorter view**
> - **title master view**

zoom

> Returns or sets the zoom setting of the specified view as a percentage of normal size. Can be a value from 10 to 400. Read/write.

zoom to fit

> **True** if the view is zoomed to fit the dimensions of the document window every time the document window is resized. This property applies only to slide view, notes page view, or master view. Read/write.

> When the value of the *zoom* property is explicitly set, the value of the *zoom to fit* property is automatically set to **false**.

Class: web options

Contains presentation-level attributes used by PowerPoint when you save or publish a presentation as a Web page or open a Web page. You can return or set attributes either at the application (global) level or at the presentation level. (Note that attribute values can be different from one presentation to another, depending on the attribute value at the time the presentation was saved.) Presentation-level attribute settings override application-level attribute settings. Application-level attributes are contained in the **default web options** object.

To return the **web options** object, use the *web options* property of the **presentation** class. The following example checks to see whether Portable Network Graphics (PNG) is allowed as an image format for presentation one. If PNG is allowed, it sets the text color for the outline pane to white and the background color for the outline and slide panes to black.

```
set objAppWebOptions to web options of presentation 1
if allow PNG of objAppWebOptions is true then
    set frame colors of objAppWebOptions to frame colors white text on black
end if
```

Properties

allow PNG

> **True** if PNG (Portable Network Graphics) is allowed as an image format when you save or publish a presentation as a Web page. **False** if PNG is not allowed as an output format. The default value is **false**. Read/write.

> If you save images in the PNG format as opposed to any other file format, you might improve the image quality or reduce the size of those image files, and therefore decrease the download time, assuming that the Web browsers you are targeting support the PNG format.

buttons type

> Determines the type of navigation buttons PowerPoint uses in a Web version of a presentation. Read/write.

> Can be one of the following:

> - **fancy** (graphic buttons)
> - **regular**
> - **text only**

encoding

Returns or sets the document encoding (code page or character set) to be used by the Web browser when you view the saved document. Read/write.

Can be one of the following:

encoding Thai
encoding Japanese ShiftJIS
encoding simplified Chinese
encoding Korean
encoding traditional Chinese
encoding little endian
encoding big endian
encoding central European
encoding Cyrillic
encoding Western
encoding Greek
encoding Turkish
encoding Hebrew
encoding Arabic
encoding Baltic
encoding Vietnamese
encoding auto detect
encoding Japanese auto detect
encoding simplified Chinese auto detect
encoding Korean auto detect
encoding traditional Chinese auto detect
encoding Cyrillic auto detect
encoding Greek auto detect
encoding Arabic auto detect
encoding ISO88591 Latin1
encoding ISO88592 central Europe
encoding ISO88593 Latin3
encoding ISO88594 Baltic
encoding ISO88595 Cyrillic
encoding ISO88596 Arabic
encoding ISO88597 Greek
encoding ISO88598 Hebrew
encoding ISO88599 Turkish
encoding ISO885915 Latin9
encoding ISO2022 Japanese no half width
   Katakana
encoding ISO2022 Japanese JISX02021984
encoding ISO2022 Japanese JISX02011989
encoding ISO2022KR
encoding ISO2022CN traditional Chinese
encoding ISO2022CN simplified Chinese
encoding Mac Roman
encoding Mac Japanese
encoding Mac traditional Chinese Big5
encoding Mac Korean
encoding Mac Greek1
encoding Mac Cyrillic

encoding EBCDIC Greek
encoding EBCDIC Hebrew
encoding EBCDIC Korean extended
encoding EBCDIC Thai
encoding EBCDIC Icelandic
encoding EBCDIC Turkish
encoding EBCDIC Russian
encoding EBCDIC Serbian Bulgarian
encoding EBCDIC Japanese Katakana
   extended and Japanese
encoding EBCDIC US Canada and Japanese
encoding EBCDIC extended and Korean
encoding EBCDIC simplified Chinese
   extended and simplified Chinese
encoding EBCDIC US Canada
   and traditional Chinese
encoding EBCDIC Japanese Latin extended
   and Japanese
encoding OEM United States
encoding OEM Greek
encoding OEM Baltic
encoding OEM multilingual LatinI
encoding OEM multilingual LatinII
encoding OEM Cyrillic
encoding OEM Turkish
encoding OEM Portuguese
encoding OEM Icelandic
encoding OEM Hebrew
encoding OEM Canadian French
encoding OEM Arabic
encoding OEM Nordic
encoding OEM CyrillicII
encoding OEM modern Greek
encoding EUC Japanese
encoding EUC Chinese simplified Chinese
encoding EUC Korean
encoding EUC Taiwanese
   traditional Chinese
encoding Devanagari
encoding Bengali
encoding Tamil
encoding Telugu
encoding Assamese
encoding Oriya
encoding Kannada
encoding Malayalam
encoding Gujarati

encoding Mac simplified Chinese GB2312	encoding Punjabi
encoding Mac Romania	encoding Arabic ASMO
encoding Mac Ukraine	encoding Arabic transparent ASMO
encoding Mac Latin2	encoding Korean Johab
encoding Mac Icelandic	encoding Taiwan CNS
encoding Mac Turkish	encoding Taiwan TCA
encoding Mac Croatia	encoding Taiwan Eten
encoding EBCDIC US Canada	encoding Taiwan IBM5550
encoding EBCDIC International	encoding Taiwan teletext
encoding EBCDIC multilingual	encoding Taiwan Wang
ROECE Latin2	encoding IA5 German
encoding EBCDIC Greek modern	encoding IA5 Swedish
encoding EBCDIC Turkish Latin5	encoding IA5 Norwegian
encoding EBCDIC Germany	encoding US ASCII
encoding EBCDIC Denmark Norway	encoding T61
encoding EBCDIC Finland Sweden	encoding ISO6937 nonspacing accent
encoding EBCDIC Italy	encoding Ext alpha lowercase
encoding EBCDIC Latin America Spain	encoding KOI8U
encoding EBCDIC United Kingdom	encoding Europa3
encoding EBCDIC Japanese	encoding HZGB simplified Chinese
Katakana extended	encoding UTF7
encoding EBCDIC France	encoding UTF8
encoding EBCDIC Arabic	

frame colors

Returns or sets the text color for the notes pane and the outline pane, and the background color for the notes pane, outline pane, and slide pane for Web presentations. Read/write.

Can be one of the following:

- **frame colors browser colors**

- **frame colors presentation scheme text color**

- **frame colors presentation scheme accent color**

- **frame colors white text on black**

- **frame colors black text on white**

include binary file

Specifies whether to include a binary version of the presentation with the Web version. Including a binary version makes it possible to open and edit the Web version in PowerPoint. Read/write.

nav bar placement

Determines the position of the navigation bar. Read/write.

Can be one of the following:

- **bar placement bottom**

- **bar placement top**

page layout

Specifies the page layout for a Web presentation. Read/write.

Can be one of the following:

- **page layout full screen**
- **page layout normal**

support IE4

Determines whether a Web presentation is optimized for Microsoft Internet Explorer 4.0 or later. Read/write.

support NN4

Determines whether a Web presentation is optimized for Netscape Navigator 4.0 or later. Read/write.

support older browsers

Determines whether a Web presentation is optimized for browsers that are older than Netscape Navigator 4.0 or Microsoft Internet Explorer 4.0. Read/write.

web page keywords

Returns or sets keywords for a presentation when it is saved as a Web page. Read/write.

Keywords are not visible to Web-page readers but are stored in meta tags to facilitate page searches.

web page title

Sets the title of a Web presentation. Read/write.

This title will appear as the title of the browser window in which the Web presentation is viewed..

Microsoft PowerPoint Suite Commands

Command: add behavior

Returns an **animation behavior** object that represents a new animation behavior.

Syntax

add behavior   effect   Required. An expression that returns an **effect** object.

> **type**   enumeration   Required. The behavior of the animation. Can be one of the following: **animation type none**, **animation type motion**, **animation type color**, **animation type scale**, **animation type rotation**, **animation type property**, **animation type command**, **animation type filter**, or **animation type set**.

Example

The following example adds a five-second animated rotation behavior to the main animation sequence on the first slide.

```
set timeMain to timeline of slide 1 of active presentation
add behavior effect 1 of main sequence of timeMain type animation type rotation
```

Command: add effect

Returns an **effect** object that represents a new animation effect added to a sequence of animation effects.

Syntax

add effect   sequence   Required. An expression that returns a **sequence** object.

  for   shape   Required. The **shape** to which the animation effect is added.

  fx   enumeration   Required. The animation effect to be applied.

    Can be one of the following:

animation type custom	**animation type flash bulb**
animation type appear	**animation type flicker**
animation type fly	**animation type grow with color**
animation type blinds	**animation type lighten**
animation type box	**animation type style emphasis**
animation type checkerboard	**animation type teeter**
animation type circle	**animation type vertical grow**
animation type crawl	**animation type wave**
animation type diamond	**animation type media play**
animation type dissolve	**animation type media pause**
animation type fade	**animation type media stop**
animation type flash once	**animation type circle path**
animation type peek	**animation type right triangle path**
animation type plus	**animation type diamond path**
animation type random bars	**animation type hexagon path**
animation type spiral	**animation type 5 point star path**
animation type split	**animation type crescent moon path**
animation type stretch	**animation type square path**
animation type strips	**animation type trapezoid path**
animation type swivel	**animation type heart path**
animation type wedge	**animation type octagon path**
animation type wheel	**animation type 6 point star path**
animation type wipe	**animation type football path**
animation type zoom	**animation type equal triangle path**
animation type random effect	**animation type parallelogram path**
animation type boomerang	**animation type pentagon path**
animation type bounce	**animation type 4 point star path**
animation type color reveal	**animation type 8 point star path**
animation type credits	**animation type teardrop path**
animation type ease in	**animation type pointy star path**
animation type float	**animation type curved square path**
animation type grow and turn	**animation type curved x path**
animation type light speed	**animation type vertical figure 8 path**
animation type pinwheel	**animation type curvy star path**
animation type rise up	**animation type loop de loop path**
animation type swish	**animation type buzzsaw path**
animation type thin line	**animation type horizontal figure 8 path**
animation type unfold	**animation type peanut path**
animation type whip	**animation type figure 8 four path**

animation type ascend	**animation type neutron path**
animation type center revolve	**animation type swoosh path**
animation type faded swivel	**animation type bean path**
animation type descend	**animation type plus path**
animation type sling	**animation type inverted triangle path**
animation type spinner	**animation type inverted square path**
animation type stretchy	**animation type left path**
animation type zip	**animation type turn right path**
animation type arc up	**animation type arc down path**
animation type fade zoom	**animation type zigzag path**
animation type glide	**animation type s curve 2 path**
animation type expand	**animation type sine wave path**
animation type flip	**animation type bounce left path**
animation type shimmer	**animation type down path**
animation type fold	**animation type turn up path**
animation type change fill color	**animation type arc up path**
animation type change font	**animation type heartbeat path**
animation type change font color	**animation type spiral right path**
animation type change font size	**animation type wave path**
animation type change font style	**animation type curvy left path**
animation type grow shrink	**animation type diagonal down right path**
animation type change line color	**animation type turn down path**
animation type spin	**animation type arc left path**
animation type transparency	**animation type funnel path**
animation type bold flash	**animation type spring path**
animation type blast	**animation type bounce right path**
animation type bold reveal	**animation type spiral left path**
animation type brush on color	**animation type diagonal up right path**
animation type brush on underline	**animation type turn up right path**
animation type color blend	**animation type arc right path**
animation type color wave	**animation type s curve 1 path**
animation type complementary color	**animation type decaying wave path**
animation type complementary color 2	**animation type curvy right path**
animation type contrasting color	**animation type stairs down path**
animation type darken	**animation type up path**
animation type desaturate	**animation type right path**

[**level** enumeration] Optional. For charts, diagrams, or text, the level to which the animation effect will be applied. The default value is **text by no levels**.

Can be one of the following:

chart all at once	**text by all levels**
chart by category	**text by fifth level**
chart by category elements	**text by first level**
chart by series	**text by fourth level**
chart by series elements	**text by second level**
text by no levels	**text by third level**

[**trigger**   enumeration]   Optional. The action that triggers the animation effect. The default value is **on page click**. Can be one of the following: **no trigger**, **on page click**, **with previous**, **after previous**, or **on shape click**.

[**index**   integer]   Optional. The position at which the effect will be placed in the collection of animation effects. The default value is -1 (added to the end).

Example

This example adds a shape to the first slide in the active presentation and adds an effect and a behavior to the shape.

```
set sldOne to slide 1 of active presentation

set shpStar to make new shape at the beginning of sldOne with properties ¬
        {auto shape type:autoshape five point star, left position:150, ¬
        top:72, width:400, height:400}

set effNew to add effect (main sequence of timeline of sldOne) for shpStar ¬
        fx animation type stretchy trigger after previous

set effBHV to add behavior effNew type animation type scale

set from x of scale effect of effBHV to 75

set from y of scale effect of effBHV to 75

set to x of scale effect of effBHV to 0

set to y of scale effect of effBHV to 0

set autoreverse of timing of effNew to true
```

Command: add sequence

Returns a new **sequence** object in a timeline.

Syntax

add sequence   timeline   Required. An expression that returns a timeline object.

[**index**   integer]   Optional. The position of the animation point or sequence in relation to other animation points or sequences. The default value is -1 which means that if the **index** argument is omitted, the new animation point or sequence is added to the end of existing animation points or sequence.

Example

This example adds a shape and an animation sequence to the first shape on the first slide in the active presentation.

```
set shpFirst to shape 1 of slide 1 of active presentation

set newSeq to add sequence (timeline of slide 1 of active presentation)

set effNew to add effect newSeq for shpFirst fx animation type blinds
```

Command: `apply template`

Applies a design template to the specified presentation.

Syntax

apply template   presentation   Required. An expression that returns a **presentation** object.

　file name   Unicode text   Required. Specifies the name of the design template.

Example

This example applies the "Zesty" design template to the active presentation.

```
apply template active presentation file name "Macintosh HD:Users:Shared:Zesty"
```

Command: `arrange windows`

Arranges all open document windows in the workspace.

Syntax

arrange windows   enumeration   Required. Specifies whether to cascade or tile the windows. Can be either of the following: **arrange cascade** or **arrange tiled**.

Example

This example arranges all open document windows.

```
arrange windows arrange cascade
```

Command: `copy object`

Copies the specified object to the Clipboard.

Syntax

copy object   slide   Required. An expression that returns a **slide** object.

Remarks

To paste the contents of the Clipboard, use the **paste object** command.

Example

This example copies slide one in the active presentation to the Clipboard.

```
copy object slide 1 of active presentation
```

Command: `cut object`

Deletes the specified object and places it on the Clipboard.

Syntax

cut object   slide   Required. An expression that returns a **slide** object.

Example

This example deletes slide one from the active presentation and places a copy of it on the Clipboard.

```
cut object slide 1 of active presentation
```

Command: exit slide show

Ends the specified slide show.

Syntax

exit slide show   slide show view   Required. An expression that returns a **slide show view** object.

Example

This example ends the slide show that's running in slide show window one.

```
exit slide show slideshow view of slide show window 1
```

Command: get color from

Returns an **RGB color** object that represents a single color in a color scheme.

Syntax

get color from   color scheme   Required. An expression that returns a **color scheme** object.

   at   enumeration   Required. The individual color in the specified color scheme. Can be one of the following: **accent1 scheme**, **accent2 scheme**, **accent3 scheme**, **background scheme**, **fill scheme**, **foreground scheme**, **scheme color unset**, **not a scheme color**, **shadow scheme**, or **title scheme**.

Example

The following example sets the title color of color scheme 2 to the title color that's defined for color scheme one.

```
set theTitle to (get color from color scheme 1 of active presentation ¬
        at title scheme)
set color for color scheme 2 of active presentation at title scheme ¬
        to color theTitle
```

Command: get text style from

Returns a **text style** object that represents one of three text styles — title text, body text, and default text — for the specified slide master.

Syntax

get text style from   master   Required. An expression that returns a **master** object.

   at   enumeration   Required. The text style you want to return. Can be one of the following: **text style default**, **text style title**, or **text style body**.

Example

The following example sets the font name and font size for level-one body text on all the slides in the active presentation.

```
set tsBody to get text style from slide master of active presentation ¬
        at text style body
set font name of font of text style level 1 of tsBody to "Arial"
set font size of font of text style level 1 of tsBody to 36
```

Command: get webPage font

Returns a **web page font** object for a particular character set.

Syntax

get webpage font   enumeration   Required. The character set from which you want to return the **web page font** object. Can be one of the following:

Arabic character set	**Korean character set**
Cyrillic character set	**Multilingual Unicode character set**
English character set	**Simplified Chinese character set**
Greek character set	**Thai character set**
Hebrew character set	**Traditional Chinese character set**
Japanese character set	**Vietnamese character set**

Example

The following example uses the **get webpage font** command to set myFont to the **web page font** object for the English character set in the active application.

```
set myFont to (get webpage font English character set)
```

Command: go to first slide

Switches to the first slide during a slide show.

Syntax

go to first slide   slide show view   Required. An expression that returns a **slide show view** object.

Example

This example switches from the current slide to the first slide in slide show window one.

```
go to first slide slideshow view of slide show window 1
```

Command: go to last slide

Switches to the last slide during a slide show.

Syntax

go to last slide   slide show view   Required. An expression that returns a **slide show view** object.

Example

This example switches from the current slide to the last slide in slide show window one.

```
go to last slide slideshow view of slide show window 1
```

Command: go to next slide

Switches to the next slide during a slide show.

Syntax

go to next slide   slide show view   Required. An expression that returns a **slide show view** object.

Example

This example switches from the current slide to the next slide in slide show window one.

```
go to next slide slideshow view of slide show window 1
```

Command: go to previous slide

Switches to the previous slide during a slide show.

Syntax

go to previous slide   slide show view   Required. An expression that returns a **slide show view** object.

Example

This example switches from the current slide to the previous slide in slide show window one.

```
go to previous slide slideshow view of slide show window 1
```

Command: go to slide

Switches to the specified slide.

Syntax

go to slide   view   Required. An expression that returns a **view** object.

  number   integer   Required. The number of the slide to switch to.

Example

This example switches from the current slide to slide three in the document window.

```
go to slide view of document window 1 number 3
```

Command: import sound file

Specifies the sound that will be played whenever the specified shape is clicked or animated or whenever the specified slide transition occurs.

Syntax

import sound file   sound effect   Required. An expression that returns a **sound effect** object.

  sound file name   Unicode text   Required. The name of the specified sound file.

Example

This example specifies that the file Dudududu.au will start to play at the transition to slide two in the active presentation and will continue to play until the next sound starts.

```
set slideTransit to slide show transition of slide 2 of active presentation
import sound file sound effect of slideTransit ¬
        sound file name "Macintosh HD:Users:Shared:Dudududu.au"
set loop sound until next of slideTransit to true
```

Command: `insert`

Inserts text at the specified location.

Syntax

insert

  the text   Unicode text   Required. The text to be inserted.

  at   location reference   Required. The location where you want to insert the text.

Example

This example appends the string ": Test version" to the end of the title on slide one in the active presentation.

```
insert the text ": Test version" at the end of text range of text frame ¬
        of shape 1 of slide 1 of active presentation
```

Command: `launch speller on`

Begins a spelling check for the specified document window. If there are errors, this command displays the **Spelling** dialog box (**Tools** menu).

Syntax

launch speller on   document window   Required. An expression that returns a **document window** object.

Example

This example begins a spelling check on the active document window.

```
launch speller on active window
```

Command: paste object

Pastes the contents of the Clipboard into the specified **view** or **presentation**. Attempting to paste an object into a view that won't accept it causes an error. For information about views and the objects you can paste into them, see the "Remarks" section.

Syntax

paste object view/presentation Required. An expression that returns a **view** or **presentation** object.

Remarks

To set the view for a window before pasting the Clipboard contents into it, use the *view type* property of the **view** object. The following table shows what you can paste into each view.

Into this view	You can paste the following from the Clipboard
Slide view or notes page view	Shapes, text, or entire slides. If you paste a slide from the Clipboard, an image of the slide will be inserted onto the slide, master, or notes page as an embedded object. If one shape is selected, the pasted text will be pasted before the shape's text; if text is selected, the pasted text will replace the selection; if anything else is selected, the pasted text will be placed in its own text frame. Pasted shapes will be added to the top of the z-order and won't replace selected shapes.
Outline view	Text or entire slides. You cannot paste shapes into outline view. A pasted slide will be inserted before the slide that contains the insertion point. Pasted text will be pasted after the insertion point if you use the **view** object or after the last slide if you use the **presentation** object.
Slide sorter view	Entire slides. You cannot paste shapes or text into slide sorter view. A pasted slide will be inserted at the insertion point or after the last slide selected in the presentation.

Example

This example copies the first shape in slide one to the Clipboard and pastes it into the view in window two. If the Clipboard contents cannot be pasted into the view in window two — for example, if you try to paste a shape into slide sorter view — this example fails.

```
copy shape shape 1 of slide 1 of active presentation
paste object view of document window 2
```

Command: play sound effect

Plays the specified sound effect.

Syntax

play sound effect sound effect Required. An expression that returns a **sound effect** object.

Example

This example plays the sound effect that's been set for the transition to slide two in the active presentation.

```
play sound effect (sound effect transition of slide show transition of ¬
        slide 2 of active presentation)
```

Command: print out

Prints the specified presentation.

Syntax

print out   presentation   Required. An expression that returns a **presentation** object.

[**from**   integer]   Optional. The number of the first page to be printed. If this argument is omitted, printing starts at the beginning of the presentation. Specifying the **fo** and **from** arguments sets the contents of the **print range** object and sets the value of the *range type* property for the presentation.

[**fo**   integer]   Optional. The number of the last page to be printed. If this argument is omitted, printing continues to the end of the presentation. Specifying the **fo** and **from** arguments sets the contents of the **print range** object and sets the value of the *range type* property for the presentation.

[**print to file**   Unicode text]   Optional. The name of the file to print to. If you specify this argument, the file is printed to a file rather than sent to a printer. If this argument is omitted, the file is sent to a printer.

[**copies**   integer]   Optional. The number of copies to be printed. If this argument is omitted, only one copy is printed. Specifying this argument sets the value of the *number of copies* property of the **print options** object.

[**collate**   Boolean]   Optional. **True** to print a complete copy of the presentation before the first page of the next copy is printed. If this argument is omitted, multiple copies are collated. Specifying this argument sets the value of the *collate* property of the **print options** object.

Example

This example prints two uncollated copies of each slide — whether visible or hidden — from slide two to slide five in the active presentation.

```
set print hidden slides of print options of active presentation to true
print out active presentation from 2 fo 5 copies 2 without collate
```

Command: quit

Quits PowerPoint. This is equivalent to clicking **Exit** on the **File** menu.

Syntax

quit   reference   Required. An expression that returns an **application** object.

Remarks

To avoid being prompted to save changes, use the **save** command to save all open presentations before calling the **quit** command.

Example

This example saves all open presentations and then quits PowerPoint.

```
repeat with w in (get presentations)
    save w
end repeat
quit
```

Command: register add in

Adds a new add-in file to the list of add-ins in the **Add-Ins** dialog box (**Tools** menu). Returns an **add in** object that represents the newly added add-in.

Syntax

register add in Unicode text Required. The full name of the file (including the path and the file name extension) that contains the add-in you want to add to the list of add-ins.

Remarks

This command doesn't load the new add-in. You must set the *loaded* property to load the add-in.

Example

This example adds MyTools.ppa to the list in the **Add-Ins** dialog box (**Tools** menu).

```
set myAddIn to register add in "Macintosh HD:Users:Shared:MyTools:MyTools.ppa"
display dialog (get name of myAddIn) & " has been added to the list"
```

Command: reset slide time

Resets the elapsed time (represented by the *slide elapsed time* property) for the slide that's currently displayed to 0 (zero).

Syntax

reset slide time slide show view Required. An expression that returns a **slide show view** object.

Example

This example resets the elapsed time for the slide that's currently displayed in slide show window one to 0 (zero).

```
reset slide time slideshow view of slide show window 1
```

Command: run slide show

Runs a slide show of the specified presentation. Returns a **slide show window** object.

Syntax

run slide show   slide show settings   Required. An expression that returns a **slide show settings** object.

Remarks

To run a custom slide show, set the *range type* property to **slide show range named slideshow**, and set the **slide show name** property to the name of the custom show you want to run.

Example

This example starts a full-screen slide show of the active presentation.

```
set mySSS to slide show settings of active presentation
set show type of mySSS to slide show type speaker
set sShow to run slide show mySSS
```

This example runs the named slide show "Quick Show."

```
set mySSS to slide show settings of active presentation
set range type of mySSS to slide show range named slideshow
set slide show name of mySSS to "Quick Show"
set sShow to run slide show mySSS
```

Command: set bullet picture

Sets the graphics file to be used for bullets in a bulleted list when the *bullet type* property of the **bullet format** object is set to **picture bullet type**.

Syntax

set bullet picture   bullet format   Required. An expression that returns a **bullet format** object of type **picture bullet type**.

   picture file   Unicode text   Required. The path and file name of a valid graphics file.

Remarks

Valid graphics files include files with the following extensions: .bmp, .emf, .eps, .gif, .jpg, .jpeg, .pct, .pict, .png, .tga, .tiff, .wmf, .fpx, .pntg, .psd, .qtif, and .sgi.

Example

This example sets the bullets in the text box specified by shape two on slide one to a bitmap picture of a blue rivet.

```
set theTF to text frame of shape 2 of slide 1 of active presentation
set theBF to bullet format of paragraph format of text range of theTF
set bullet type of theBF to picture bullet type
set bullet picture theBF picture file "Macintosh HD:Users:Shared:Blue Rivets"
```

Command: `set color for`

Sets an individual color in a color scheme.

Syntax

set color for   color scheme   Required. An expression that returns a **color scheme** object.

> **at** enumeration   Required. The individual color in the specified color scheme. Can be one of the following: **accent1 scheme**, **accent2 scheme**, **accent3 scheme**, **background scheme**, **fill scheme**, **foreground scheme**, **scheme color unset**, **not a scheme color**, **shadow scheme**, or **title scheme**.

> **to color**   RGB color   Required. The color to set.

Example

The following example sets the title color of color scheme 2 to the title color that's defined for color scheme one.

```
set theTitle to (get color from color scheme 1 of active presentation ¬
        at title scheme)
set color for color scheme 2 of active presentation at title scheme ¬
        to color theTitle
```

Command: `update links`

Updates linked OLE objects in the specified presentation.

Syntax

update links   presentation   Required. An expression that returns a **presentation** object.

Example

This example updates all OLE links in the active presentation.

```
update links active presentation
```

Drawing Suite

Drawing Suite Classes

Class: `callout`

Plural

callouts

Represents a line callout. The **callout** class inherits all the properties of the **shape** class.

Use **callout** *index*, where *index* is the name or the index number, to return a single **callout** object. To set the formatting options for a **callout**, use the **callout format** class.

This example adds a callout to a newly created slide and then sets the callout angle to 45 degrees.

```
set thePres to active presentation
set theSlide to make new slide at the beginning of thePres with properties ¬
        {layout: slide layout blank}
set myCall to make new callout at the beginning of theSlide with properties ¬
        {callout type:callout two, left position:72, top:36, width:25, ¬
        height:25}
set angle of callout format of myCall to angle45
```

Properties

<Inheritance> **shape**

> Inherits the properties and elements of the **shape** class.

callout format

> Returns a **callout format** object that contains callout formatting properties for the specified shape. Read-only.

callout Type

> Returns the callout type. Read only.

> Can be one of the following:

Value	Description
callout unset	
callout one	A single-segment callout line that can be either horizontal or vertical
callout two	A single-segment callout line that rotates freely
callout three	A two-segment line
callout four	A three-segment line

Class: callout format

Contains properties that apply to line callouts.

Use the *callout format* property to return a **callout format** object. The following example specifies these attributes of shape three (a line callout) on `theSlide`: the callout will have a vertical accent bar that separates the text from the callout line; the angle between the callout line and the side of the callout text box will be 30 degrees; there will be no border around the callout text; the callout line will be attached to the top of the callout text box; and the callout line will contain two segments. For this example to work, shape three must be a callout.

```
set theSlide to slide 1 of active presentation
set theCF to callout format of shape 3 of theSlide
set accent of theCF to true
set angle of theCF to angle30
set border of theCF to false
preset drop theCF drop type drop top
set callout type of theCF to callout three
```

Properties

accent

True if a vertical accent bar separates the callout text from the callout line. Read/write.

angle

Returns or sets the angle of the callout line. If the callout line contains more than one line segment, this property returns or sets the angle of the segment that is farthest from the callout text box. Read/write.

Can be one of the following:

- **angle30**

- **angle45**

- **angle60**

- **angle90**

- **angle automatic**

- **angle unset**

If you set the value of this property to anything other than **angle automatic**, the callout line maintains a fixed angle as you drag the callout.

auto attach

True if the place where the callout line attaches to the callout text box changes depending on whether the origin of the callout line (where the callout points to) is to the left or right of the callout text box. Read/write.

When the value of this property is **true**, the drop value (the vertical distance from the edge of the callout text box to the place where the callout line attaches) is measured from the top of the text box when the text box is to the right of the origin, and it's measured from the bottom of the text box when the text box is to the left of the origin. When the value of this property is **false**, the drop value is always measured from the top of the text box, regardless of the relative positions of the text box and the origin. To set the drop value, use the **custom drop** command, and use the *drop* property to return the drop value.

Setting this property affects a callout only if it has an explicitly set drop value, that is, if the value of the *drop type* property is **drop custom**. By default, callouts have explicitly set drop values when they're created.

auto length

True if the first segment of the callout line (the segment attached to the text callout box) is scaled automatically whenever the callout is moved. **False** if the first segment of the callout retains the fixed length specified by the *callout format length* property whenever the callout is moved. Applies only to callouts whose lines consist of more than one segment (types **callout three** and **callout four**). Read-only.

This property is read-only. To set this property to **true**, use the **automatic length** command. To set this property to **false**, use the **custom length** command.

border

Determines whether the text in the specified callout is surrounded by a border. Read/write

callout format length

When the *auto length* property of the specified callout is set to **false**, the *callout format length* property returns the length (in points) of the first segment of the callout line (the segment attached to the text callout box). Applies only to callouts whose lines consist of more than one segment (types **callout three** and **callout four**). Read-only.

This property is read-only. To set the value of this property, use the **custom length** command.

callout has border

True if the text in the specified callout is surrounded by a border. Read/write.

callout type

Returns or sets the callout type. Read only.

Can be one of the following:

- **callout unset**
- **callout one**
- **callout two**
- **callout three**
- **callout four**

drop

For callouts with an explicitly set drop value, this property returns the vertical distance (in points) from the edge of the text bounding box to the place where the callout line attaches to the text box. This distance is measured from the top of the text box unless the *auto attach* property is set to **true** and the text box is to the left of the origin of the callout line (the place that the callout points to), in which case the drop distance is measured from the bottom of the text box. Read-only.

To set the value of this property, use the **custom drop** command.

The value of this property accurately reflects the position of the callout line attachment to the text box only if the callout has an explicitly set drop value, that is, if the value of the *drop type* property is **drop custom**. To set the *drop type* property to **drop custom**, use the **preset drop** command.

drop type

> Returns a value that indicates where the callout line attaches to the callout text box. Read-only.
>
> Can be one of the following:

- **drop bottom**
- **drop center**
- **drop custom**
- **drop unset**
- **drop top**

> If the callout drop type is **drop custom**, the values of the *drop* and *auto attach* properties and the relative positions of the callout text box and callout line origin (the place that the callout points to) are used to determine where the callout line attaches to the text box.
>
> This property is read-only. To set the value of this property, use the **preset drop** command.

gap

> Returns or sets the horizontal distance (in points) between the end of the callout line and the text bounding box. Read/write.

Class: comment

Represents a comment on a given slide.

Use **comment** *index*, where *index* is the number of the comment to access a single comment on a slide. This example displays the content of the first comment on the first slide. If there are no comments, it displays a message stating such.

```
set allComments to (get comments of slide 1 of active presentation)
if count of allComments > 0 then
    display dialog "Comment: " & name of item 1 of allComments
else
    display dialog "There are no comments on this slide."
end if
```

Properties

<Inheritance> **shape**

> Inherits the properties and elements of the **shape** class.

Class: connector

Plural

connectors

Represents a connector.

Use **connector** *index*, where *index* is the number of the **connector** to access a single comment on a slide. Use the **make** command to create a new **connector** object. When a **connector** is added, it's not connected to anything. To attach the beginning and end of a connector to other shapes in the document, use the **begin connect** and **end connect** commands.

This example adds two shapes to `myPres` and connects them with a curved connector. Note that when you attach the connector to the shapes, the size and position of the connector are automatically adjusted; therefore, the position and dimensions you specify when adding the connector are irrelevant (dimensions must be nonzero).

```
set myPres to slide 1 of active presentation

set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}

set shpTri to make new shape at the end of myPres with properties ¬
        {auto Shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}

set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}

begin connect shpCon connected shape shpTrap connection site 1

end connect shpCon connected shape shpTri connection site 1

reroute connections shpCon
```

When you attach a connector to a shape, the size and position of the connector are automatically adjusted, if necessary. Therefore, if you're going to attach a connector to other shapes, the position and dimensions you specify when adding the connector are irrelevant.

Properties

connector format

Returns a **connector format** object that contains formatting properties for the specified connector. Read-only.

connector type

Returns the type of connector. Read-only.

Can be one of the following:

- **connector type unset**

- **straight**

- **elbow**

- **curve**

Class: connector format

Contains formatting properties that apply to connectors. A connector is a line that attaches two other shapes at points called connection sites. If you rearrange shapes that are connected, the geometry of the connector will automatically be adjusted so that the shapes remain connected.

To return a **connector format** object, use the *connector format* property. To attach the ends of the connector to other shapes in the document, use the **begin connect** and **end connect** commands. To automatically find the shortest path between the two shapes connected by the connector, use the **reroute connections** command.

Note that you assign a size and a position when you create a **connector** object, but the size and position are automatically adjusted when you attach the beginning and end of the connector to other shapes. Therefore, if you intend to attach a connector to other shapes, the initial size and position you specify are irrelevant. Likewise, you specify which connection sites on a shape to attach the connector to when you attach the connector, but using the **reroute connections** command after the connector is attached may change which connection sites the connector attaches to, making your original choice of connection sites irrelevant.

The following example adds two shapes to `myPres` and connects them with a curved connector.

```
set myPres to slide 1 of active presentation
set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}
set shpTri to make new shape at the end of myPres with properties ¬
        {auto shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}
set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}
begin connect of shpCon connected shape shpTrap connection site 1
end connect shpCon connected shape shpTri connection site 1
reroute connections shpCon
```

Properties

begin connected

> **True** if the beginning of the specified connector is connected to a shape. Read-only.

begin connected shape

> Returns a **shape** object that represents the shape that the beginning of the specified connector is attached to. Read-only.

> **Note**   If the beginning of the specified connector isn't attached to a shape, this property generates an error.

begin connection site

>Returns an integer that specifies the connection site that the beginning of a connector is connected to. Read-only.

>**Note** If the beginning of the specified connector isn't attached to a shape, this property generates an error.

connector type

>Returns or sets the type of connector. Read/write.

>Can be one of the following:

>>- **connector type unset**
>>- **straight**
>>- **elbow**
>>- **curve**

end connected

>**True** if the end of the specified connector is connected to a shape. Read-only.

end connected shape

>Returns a **shape** object that represents the shape that the end of the specified connector is attached to. Read-only.

>**Note** If the end of the specified connector isn't attached to a shape, this property generates an error.

end connection site

>Returns an integer that specifies the connection site that the end of a connector is connected to. Read-only.

>**Note** If the end of the specified connector isn't attached to a shape, this property generates an error.

Class: fill format

Represents fill formatting for a shape. A shape can have a solid, gradient, texture, pattern, picture, or semi-transparent fill.

Use the *fill* property to return a shape's **fill format** object. The following example adds a rectangle to myPres and then sets the gradient and color for the rectangle's fill.

```
set myPres to slide 1 of active presentation
set shpRect to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape rectangle, left position:90, top:90, ¬
        width:90, height:80}
set fore color of fill of shpRect to ({0, 128, 128} as RGB color)
one color gradient fill of shpRect style horizontal gradient variant 1 ¬
        degree 1
```

Remarks

Many of the properties of the **fill format** object are read-only. To set one of these properties, you have to apply the corresponding command.

Properties

back color

> Returns or sets an RGB color that represents the background color for the specified fill format. Read/write.

fill format type

> Returns the shape fill format type. Read-only.

> Can be one of the following

> - **fill unset**
> - **fill solid**
> - **fill patterned**
> - **fill gradient**
> - **fill textured**
> - **fill background**
> - **fill picture**

fore color

> Returns or sets an RGB color that represents the foreground color for the fill format. Read-only.

gradient color type

> Returns the gradient color type for the specified fill. Read-only.

> Can be one of the following:

> - **gradient type unset**
> - **single shade gradient type**
> - **two colors gradient type**
> - **preset colors gradient type**

> To set the gradient type for the fill format, use the **one color gradient**, **preset gradient**, or **two color gradient** command.

gradient degree

Returns a value that indicates how dark or light a one-color gradient **fill format** is. A value of 0 (zero) means that black is mixed in with the shape's foreground color to form the gradient; a value of 1 means that white is mixed in; and values between 0 and 1 mean that a darker or lighter shade of the foreground color is mixed in. Read-only.

To set the gradient degree for the fill format, use the **one color gradient** command.

gradient style

Returns the gradient style for the specified fill. Read-only.

Can be one of the following:

- **gradient unset**

- **horizontal gradient**

- **vertical gradient**

- **diagonal up gradient**

- **diagonal down gradient**

- **from corner gradient**

- **from center gradient**

- **from title gradient**

To set the gradient style for the fill, use the **one color gradient** or **two color gradient** command.

Note   Attempting to return this property for a fill that doesn't have a gradient generates an error. To determine whether the fill has a gradient, use the *fill type* property.

gradient variant

Returns the gradient variant for the specified fill format as an integer value from 1 to 4 for most gradient fills. If the gradient style is **from center gradient**, this property returns either 1 or 2. The values for this property correspond to the gradient variants (numbered from left to right and from top to bottom) on the **Gradient** tab in the **Fill Effects** dialog box. Read-only.

To set the gradient variant for the fill format, use the **one color gradient** or **two color gradient** command.

Drawing Suite

pattern

Returns a value that represents the pattern applied to the specified fill format. Read-only.

Can be one of the following:

five percent pattern	**large checker board pattern**
ten percent pattern	**large confetti pattern**
twenty percent pattern	**large grid pattern**
twenty five percent pattern	**light downward diagonal pattern**
thirty percent pattern	**light horizontal pattern**
forty percent pattern	**light upward diagonal pattern**
fifty percent pattern	**light vertical pattern**
sixty percent pattern	**unset pattern**
seventy percent pattern	**narrow horizontal pattern**
seventy five percent pattern	**narrow vertical pattern**
eighty percent pattern	**outlined diamond pattern**
ninety percent pattern	**plaid pattern**
dark downward diagonal pattern	**shingle pattern**
dark horizontal pattern	**small checker board pattern**
dark upward diagonal pattern	**small confetti pattern**
dark vertical pattern	**small grid pattern**
dashed downward diagonal pattern	**solid diamond pattern**
dashed horizontal pattern	**sphere pattern**
dashed upward diagonal pattern	**trellis pattern**
dashed vertical pattern	**wave pattern**
diagonal brick pattern	**weave pattern**
divot pattern	**wide downward diagonal pattern**
dotted diamond pattern	**wide upward diagonal pattern**
dotted grid pattern	**zig zag pattern**
horizontal brick pattern	

To set the pattern for the fill format, use the **patterned** command.

To set the colors used in the pattern, use the *back color* and *fore color* properties.

preset gradient type

Returns the preset gradient type for the specified fill format. Read-only.

Can be one of the following:

gradient brass	**gradient mahogany**
gradient calm water	**gradient moss**
gradient chrome	**gradient nightfall**
gradient chrome2	**gradient ocean**
gradient daybreak	**gradient parchment**
gradient desert	**gradient peacock**
gradient early sunset	**gradient rainbow**
gradient fire	**gradient rainbow2**
gradient fog	**gradient sapphire**
gradient gold	**gradient silver**
gradient gold2	**gradient wheat**
gradient horizon	**preset gradient unset**
gradient late sunset	

To set the preset gradient type for the fill format, use the **preset gradient** command.

preset texture

Returns the preset texture for the specified fill format. Read-only.

Can be one of the following:

preset texture unset	**texture paper bag**
texture blue tissue paper	**texture papyrus**
texture bouquet	**texture parchment**
texture brown marble	**texture pink tissue paper**
texture canvas	**texture purple mesh**
texture cork	**texture recycled paper**
texture denim	**texture sand**
texture fish fossil	**texture stationery**
texture granite	**texture walnut**
texture green marble	**texture water droplets**
texture medium wood	**texture white marble**
texture newsprint	**texture woven mat**
texture oak	

To set the preset texture for the fill format, use the **preset textured** command.

texture name

Returns the name of the custom texture file for the specified fill format. Read-only.

To set the texture file for the fill format, use the **user textured** command.

transparency

Returns or sets the degree of transparency of the specified fill format as a value between 0.0 (opaque) and 1.0 (clear). Read/write.

The value of this property affects the appearance of solid-colored fills and lines only; it has no effect on the appearance of patterned lines or patterned, gradient, picture, or textured fills.

visible

True if the specified object, or the formatting applied to it, is visible. Read/write.

Class: line format

Represents line and arrowhead formatting. For a line, the **line format** object contains formatting information for the line itself; for a shape with a border, this object contains formatting information for the shape's border.

Use the *line format* property to return a **line format** object. The following example adds a a blue dashed line to myPres. There's a short narrow oval at the line's starting point and a long wide triangle at its end point.

```
set myPres to slide 1 of active presentation
set shpLine to make new line shape at the beginning of myPres with properties ¬
        {left position:100, top:100, width:200, height:300}
set lfshpLine to line format of shpLine
set dash style of lfshpLine  to line dash style dash dot dot
set fore color of lfshpLine  to ({50, 0, 128} as RGB color)
set begin arrow head length of lfshpLine to short arrowhead
set begin arrowhead style of lfshpLine to oval arrowhead
set begin arrowhead width of lfshpLine to narrow width arrowhead
set end arrowhead length of lfshpLine to long arrowhead
set end arrowhead style of lfshpLine to triangle arrowhead
set end arrowhead width of lfshpLine to wide arrowhead
```

Properties

back color

> Returns or sets an RGB color that represents the background color for the specified line format. Read/write.

begin arrowhead length

> Returns or sets the length of the arrowhead at the beginning of the specified line. Read/write.
>
> Can be one of the following:

- **arrowhead length unset**
- **short arrowhead**
- **medium arrowhead**
- **long arrowhead**

begin arrowhead style

Returns or sets the style of the arrowhead at the beginning of the specified line. Read/write.

Can be one of the following:

- **arrowhead style unset**
- **no arrowhead**
- **triangle arrowhead**
- **open_arrowhead**
- **stealth arrowhead**
- **diamond arrowhead**
- **oval arrowhead**

begin arrowhead width

Returns or sets the width of the arrowhead at the beginning of the specified line. Read/write.

Can be one of the following:

- **arrowhead width unset**
- **narrow width arrowhead**
- **medium width arrowhead**
- **wide arrowhead**

dash style

Returns or sets the dash style for the specified line. Read/write.

Can be one of the following:

- **line dash style unset**
- **line dash style solid**
- **line dash style square dot**
- **line dash style round dot**
- **line dash style dash**
- **line dash style dash dot dot**
- **line dash style long dash**
- **line dash style dash dot**
- **line dash style long dash dot**

end arrowhead length

Returns or sets the length of the arrowhead at the end of the specified line. Read/write.

Can be one of the following:

- **arrowhead length unset**
- **short arrowhead**
- **medium arrowhead**
- **long arrowhead**

end arrowhead style

Returns or sets the style of the arrowhead at the end of the specified line. Read/write.

Can be one of the following:

- **arrowhead style unset**
- **no arrowhead**
- **triangle arrowhead**
- **open_arrowhead**
- **stealth arrowhead**
- **diamond arrowhead**
- **oval arrowhead**

end arrowhead width

Returns or sets the width of the arrowhead at the end of the specified line. Read/write.

Can be one of the following:

- **arrowhead width unset**
- **narrow width arrowhead**
- **medium width arrowhead**
- **wide arrowhead**

fore color

Returns or sets an RGB color that represents the foreground color for the line format. Read/write.

line format patterned

Returns or sets a value that represents the pattern applied to the specified line format. Read/write.

Can be one of the following:

five percent pattern	**large checker board pattern**
ten percent pattern	**large confetti pattern**
twenty percent pattern	**large grid pattern**
twenty five percent pattern	**light downward diagonal pattern**
thirty percent pattern	**light horizontal pattern**
forty percent pattern	**light upward diagonal pattern**
fifty percent pattern	**light vertical pattern**
sixty percent pattern	**unset pattern**
seventy percent pattern	**narrow horizontal pattern**
seventy five percent pattern	**narrow vertical pattern**
eighty percent pattern	**outlined diamond pattern**
ninety percent pattern	**plaid pattern**
dark downward diagonal pattern	**shingle pattern**
dark horizontal pattern	**small checker board pattern**
dark upward diagonal pattern	**small confetti pattern**
dark vertical pattern	**small grid pattern**
dashed downward diagonal pattern	**solid diamond pattern**
dashed horizontal pattern	**sphere pattern**
dashed upward diagonal pattern	**trellis pattern**
dashed vertical pattern	**wave pattern**
diagonal brick pattern	**weave pattern**
divot pattern	**wide downward diagonal pattern**
dotted diamond pattern	**wide upward diagonal pattern**
dotted grid pattern	**zig zag pattern**
horizontal brick pattern	

line style

Returns or sets the line style for the fill format object. Read/write.

Can be one of the following:

- **line style unset**

- **single line**

- **thin thin line**

- **thin thick line**

- **thick thin line**

- **thick between thin line**

line weight

Returns or sets the thickness (in points) of the specified line. Read/write.

transparency

> Returns or sets the degree of transparency of the specified line as a value between 0.0 (opaque) and 1.0 (clear). Read/write.

> The value of this property affects the appearance of solid-colored lines only; it has no effect on the appearance of patterned lines.

Class: line shape

Plural

line shapes

Represents a line drawn on a slide. The **line shape** class inherits all the properties of the **shape** class.

Use **line shape** *index*, where *index* is the name or the index number, to return a single **line shape** object.

This example adds a dashed green line to a new document.

```
set myPres to active presentation
set myLine to make new line shape at the beginning of myPres with properties ¬
        {begin line X:100, begin line Y:100, end line X:60, end line Y:20}
set dash style of line format of myLine to line dash style dash
set fore color of line format of myLine to ({0, 128, 0} as RGB color)
```

This example formats the first line on slide 1 of the active presentation as a red arrow.

```
set begin arrowhead style of line format of line shape 1 of slide 1 of ¬
        active presentation to no arrowhead
set end arrowhead style of line format of line shape 1 of slide 1 of ¬
        active presentation to triangle arrowhead
set fore color of line format of line shape 1 of slide 1 of ¬
        active presentation to ({128, 0, 0} as RGB color)
```

Properties

<Inheritance> **shape**

> Inherits the properties and elements of the **shape** class.

begin line X

> Returns or sets the starting X coordinate for the line shape. Read/write.

begin line Y

> Returns or sets the starting Y coordinate for the line shape. Read/write.

end line X

> Returns or sets the ending X coordinate for the line shape. Read/write.

end line Y

> Returns or sets the ending Y coordinate for the line shape. Read/write.

Class: link format

Contains properties that apply to linked OLE objects.

Use the *link format* property to return a **link format** object.

Properties

auto update

Returns or sets the way the link will be updated. Read/write.

Can be one of the following:

- **update option manual**

- **update option unset**

source full name

Returns or sets the name and path of the source file for the linked OLE object. Read/write.

Class: picture

Plural

pictures

Represents a picture shape.

Use **picture** *index*, where *index* is the picture name or the index number, to return a **picture** object. The following example creates a new picture on slide 1 from the file "picture.jpg."

```
set thePres to active presentation
make new picture at the beginning of slide 1 of thePres with properties ¬
        {file name:"Macintosh HD:Shared:Pictures:picture.jpg", ¬
        top:100, left position:100, height:300, width:300}
```

Properties

<Inheritance> **shape**

Inherits the properties and elements of the **shape** class.

file name

Returns or sets the URL (on the intranet or the Web) or path (local or network) to the location where the specified source object was saved. Read/write.

The *file name* property generates an error if a folder in the specified path doesn't exist.

link to file

True if the picture is linked to the file. Read-only.

picture format

Returns a **picture format** object that contains picture formatting properties. Read-only.

save with document

True if the picture should be saved with the document. Read-only.

Class: picture format

Contains properties that apply to pictures.

Use the *picture format* property to return a **picture format** object. The following example sets the brightness, contrast, and color transformation for picture one on `myPres` and crops 18 points off the bottom of the shape.

```
set myPres to slide 1 of active presentation
set brightness of picture format of picture 1 of myPres to 0.3
set contrast of picture format of picture 1 of myPres to 0.7
set color type of picture format of picture 1 of myPres to ¬
        picture color gray scale
set crop bottom of picture format of picture 1 of myPres to 18
```

Properties

brightness

> Returns or sets the brightness of the specified picture. The value for this property must be a number from 0.0 (dimmest) to 1.0 (brightest). Read/write.

color type

> Returns or sets the type of color transformations applied to the specified picture. Read/write.
>
> Can be one of the following:
>
> - **picture color type unset**
> - **picture color automatic**
> - **picture color gray scale**
> - **picture color black and white**
> - **picture color watermark**

contrast

> Returns or sets the contrast for the specified picture. The value for this property must be a number from 0.0 (the least contrast) to 1.0 (the greatest contrast). Read/write.

crop bottom

> Returns or sets the number of points that are cropped off the bottom of the specified picture. Read/write.

crop left

> Returns or sets the number of points that are cropped off the left side of the specified picture. Read/write.

crop right

> Returns or sets the number of points that are cropped off the right side of the specified picture. Read/write.

crop top

> Returns or sets the number of points that are cropped off the top of the specified picture. Read/write.

transparency color

> Returns or sets the transparent color for the specified picture as an RGB value. For this property to take effect, the *transparent background* property must be set to **true**. Read/write.

transparent background

> **True** if the parts of the picture that are the color defined as the transparent color appear transparent. To set the transparent color, use the *transparency color* property. Read/write.

Class: place holder

Plural

place holders

Represents a placeholder shape on a slide. Each **place holder** object represents a placeholder for text, a chart, a table, an organizational chart, or some other type of object. If the slide has a title, the title is the first placeholder in the placeholders list.

Use **place holder** *index*, where *index* is the placeholder index number, to return a **place holder** object that represents a single placeholder. The following example adds a new slide with a Bulleted List slide layout to the beginning of the presentation, sets the text for the title, and then adds two paragraphs to the text placeholder.

```
set sObj to make new slide at the beginning of active presentation ¬
        with properties {layout:slide layout text slide}
set content of text range of text frame of place holder 1 of sObj ¬
        to "This is the title text"
set content of text range of text frame of place holder 2 of sObj ¬
        to "Item 1" & return & "Item 2"
```

You can delete individual placeholders by using the **delete** command, and you can restore deleted placeholders by using the **make** command, but you cannot add any more placeholders to a slide than it had when it was created. To change the number of placeholders on a given slide, set the *layout* property.

Properties

place holder format

> Returns a **placeholder format** object that contains the properties that are unique to placeholders. Read-only.

placeholder type

> Returns the placeholder type for the specified placeholder. Read-only.

> Can be one of the following:

placeholder type unset	**placeholder type media clip placeholder**
placeholder type title placeholder	**placeholder type object placeholder**
placeholder type bitmap placeholder	**placeholder type org chart placeholder**
placeholder type body placeholder	**placeholder type slide number placeholder**
placeholder type center title placeholder	**placeholder type subtitle placeholder**
placeholder type chart placeholder	**placeholder type table placeholder**
placeholder type date placeholder	**placeholder type vertical body placeholder**
placeholder type footer placeholder	**placeholder type vertical title placeholder**
placeholder type header placeholder	

Class: placeholder format

Contains properties that apply specifically to placeholders.

Use the *placeholder format* property to return a **placeholder format** object. The following example adds text to placeholder one on slide one in the active presentation if that placeholder exists and is a horizontal title placeholder.

```
set allPlace to place holders of slide 1 of active presentation

if (count of place holders of allPlace) > 0 then

    if (placeholder type of item 1 of allPlace is placeholder type title ¬

        placeholder) or (placeholder type of item 1 of allPlace is ¬

        placeholder type center title placeholder) then

        set content of text range of text frame of item 1 of allPlace to ¬

        "This is the title"

    else

        display dialog "There is no horizontal title on this slide"

    end if

end if
```

Properties

placeholder type

> Returns the placeholder type for the specified placeholder. Read-only.

> Can be one of the following:

placeholder type unset	**placeholder type media clip placeholder**
placeholder type title placeholder	**placeholder type object placeholder**
placeholder type bitmap placeholder	**placeholder type org chart placeholder**
placeholder type body placeholder	**placeholder type slide number placeholder**
placeholder type center title placeholder	**placeholder type subtitle placeholder**
placeholder type chart placeholder	**placeholder type table placeholder**
placeholder type date placeholder	**placeholder type vertical body placeholder**
placeholder type footer placeholder	**placeholder type vertical title placeholder**
placeholder type header placeholder	

Class: shadow format

Represents shadow formatting for a shape.

Use the *shadow format* property to return a **shadow format** object. The following example adds a shadowed rectangle to myPres. The semitransparent blue shadow is offset 5 points to the right of the rectangle and 3 points above it.

```
set myPres to slide 1 of active presentation
set shpRect to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape rectangle, left position:50, top:50, ¬
        width:100, height:200}
set fore color of shadow format of shpRect to ({0, 0, 128} as RGB color)
set X offset of shadow format of shpRect to 5
set y offset of shadow format of shpRect to -3
set transparency of shadow format of shpRect to 0.5
set visible of shadow format of shpRect to true
```

Properties

X offset

> Returns or sets the horizontal offset (in points) of the shadow from the specified shape. A positive value offsets the shadow to the right of the shape; a negative value offsets it to the left. Read/write.

Y offset

> Returns or sets the vertical offset (in points) of the shadow from the specified shape. A positive value offsets the shadow below the shape; a negative value offsets it above the shape. Read/write.

fore color

> Returns or sets an RGB color that represents the foreground color for the line format. Read-only.

obscured

> **True** if the shadow of the specified shape appears filled in and is obscured by the shape, even if the shape has no fill. **False** if the shadow has no fill and the outline of the shadow is visible through the shape if the shape has no fill. Read/write.

shadow type

> Returns or sets the shape shadow type. Read/write.
>
> Can be one of the following:

shadow unset	**shadow7**	**shadow14**
shadow1	**shadow8**	**shadow15**
shadow2	**shadow9**	**shadow16**
shadow3	**shadow10**	**shadow17**
shadow4	**shadow11**	**shadow18**
shadow5	**shadow12**	**shadow19**
shadow6	**shadow13**	**shadow20**

transparency

> Returns or sets the degree of transparency of the specified shadow as a value between 0.0 (opaque) and 1.0 (clear). Read/write.

visible

> **True** if the **shadow format** object, or the formatting applied to it, is visible. Read/write.

Class: shape

Plural
shapes

Elements
shape

callout

connector

picture

line shape

placeholder

word art

text box

comment

shape table

Represents an object in the drawing layer, such as an AutoShape, freeform, OLE object, or picture. The **shapes** list contains all the shapes on a slide.

Use **shape** *index*, where *index* is the shape name or the index number, to return a **shape** object that represents a shape on a slide. Shapes on slides are numbered sequentially, regardless of the shape type. The following example horizontally flips shape one and the shape named "Rectangle 1" on myPres.

```
set myPres to slide 1 of active presentation
```

```
flip shape 1 of myPres direction flip horizontal
```

```
flip shape "Rectangle 1" of myPres direction flip horizontal
```

Each shape is assigned a default name when it is created. To give the shape a more meaningful name, use the **name** property. The following example adds a rectangle to myPres, gives it the name "Red Square," and then sets its foreground color and line style.

```
set myPres to slide 1 of active presentation
```

```
set shpRect to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape rectangle, left position:144, top:144, ¬
        width:72, height:72}
```

```
set name of shpRect to "Red Square"
```

```
set fore color of fill of shpRect to ({255, 0, 0} as RGB color)
```

```
set dash style of line format of shpRect to line dash style dash dot
```

Properties

animation settings

Returns an **animation settings** object that represents all the special effects you can apply to the animation of the specified shape.

auto shape type

Returns or sets the type of AutoShape. Read/write.

Can be one of the following:

autoshape rectangle	**autoshape parallelogram**
autoshape trapezoid	**autoshape diamond**
autoshape rounded rectangle	**autoshape octagon**
autoshape isosceles triangle	**autoshape right triangle**
autoshape oval	**autoshape hexagon**
autoshape cross	**autoshape regular pentagon**
autoshape can	**autoshape cube**
autoshape bevel	**autoshape folded corner**
autoshape smiley face	**autoshape donut**
autoshape no symbol	**autoshape block arc**
autoshape heart	**autoshape lightning bolt**
autoshape sun	**autoshape moon**
autoshape arc	**autoshape double bracket**
autoshape double brace	**autoshape plaque**
autoshape left bracket	**autoshape right bracket**
autoshape left brace	**autoshape right brace**
autoshape right arrow	**autoshape left arrow**
autoshape up arrow	**autoshape down arrow**
autoshape left right arrow	**autoshape up down arrow**
autoshape quad arrow	**autoshape left right up arrow**
autoshape bent arrow	**autoshape U turn arrow**
autoshape left up arrow	**autoshape dent up arrow**
autoshape curved right arrow	**autoshape curved left arrow**
autoshape curved up arrow	**autoshape curved down arrow**
autoshape striped right arrow	**autoshape notched right arrow**
autoshape pentagon	**autoshape chevron**
autoshape right arrow callout	**autoshape left arrow callout**
autoshape up arrow callout	**autoshape down arrow callout**
autoshape left right arrow callout	**autoshape up down arrow callout**
autoshape quad arrow callout	**autoshape circular arrow**
autoshape flowchart process	**autoshape flowchart alternate process**
autoshape flowchart decision	**autoshape flowchart data**
autoshape flowchart predefined process	**autoshape flowchart internal storage**
autoshape flowchart document	**autoshape flowchart multi document**
autoshape flowchart terminator	**autoshape flowchart preparation**
autoshape flowchart manual input	**autoshape flowchart manual operation**
autoshape flowchart connector	**autoshape flowchart offpage connector**
autoshape flowchart card	**autoshape flowchart punched tape**
autoshape flowchart summing junction	**autoshape flowchart or**
autoshape flowchart collate	**autoshape flowchart sort**
autoshape flowchart extract	**autoshape flowchart merge**
autoshape flowchart stored data	**autoshape flowchart delay**
autoshape flowchart sequential access storage	**autoshape flowchart magnetic disk**

autoshape flowchart direct access storage   **autoshape flowchart display**
autoshape explosion one   **autoshape explosion two**
autoshape four point star   **autoshape five point star**
autoshape eight point star   **autoshape sixteen point star**
autoshape twenty four point star   **autoshape thirty two point star**
autoshape up ribbon   **autoshape down ribbon**
autoshape curved up ribbon   **autoshape curved down ribbon**
autoshape vertical scroll   **autoshape horizontal scroll**
autoshape wave   **autoshape double wave**
autoshape rectangular callout   **autoshape rounded rectangular callout**
autoshape oval callout   **autoshape cloud callout**
autoshape line callout one   **autoshape line callout two**
autoshape line callout three   **autoshape line callout four**
autoshape line callout one accent bar   **autoshape line callout two accent bar**
autoshape line callout three accent bar   **autoshape line callout four accent bar**
autoshape line callout one no border   **autoshape line callout two no border**
autoshape line callout three no border   **autoshape line callout four no border**
autoshape callout one border and accent bar **autoshape callout two border and accent bar**
autoshape callout three border and accent bar **autoshape callout four border and accent bar**
autoshape action button custom   **autoshape action button home**
autoshape action button help   **autoshape action button information**
autoshape action button back or previous   **autoshape action button forward or next**
autoshape action button beginning   **autoshape action button end**
autoshape action button return   **autoshape action button document**
autoshape action button sound   **autoshape action button movie**
autoshape balloon

black and white mode

Returns or sets a value that indicates how the specified shape appears when the document is viewed in black-and-white mode. Read/write.

Can be one of the following:

black and white mode automatic   **black and white mode high contrast**
black and white mode black   **black and white mode inverse gray scale**
black and white mode black text and line **black and white mode light gray scale**
black and white mode dont show   **black and white mode unset**
black and white mode gray outline   **black and white mode white**
black and white mode gray scale

connection site count

Returns the number of connection sites on the specified shape. Read-only.

fill

Returns a **fill format** object that contains fill formatting properties for the specified shape. Read-only.

has connector

True if the specified shape is a connector. Read-only.

has table

True if the specified shape is a table. Read-only.

has text frame

> **True** if the specified shape has a text frame and can therefore contain text. Read-only.

height

> Returns or sets the height of the specified shape. Read/write.

horizontal flip

> **True** if the specified shape is flipped around the horizontal axis. Read-only.

left position

> Returns or sets the horizontal position (in points) of the specified shape or shape range. Read/write.

line format

> Returns a **line format** object that contains line formatting properties for the specified shape. (For a line, the **line format** object represents the line itself; for a shape with a border, the **line format** object represents the border.) Read-only.

link format

> Returns a **link format** object that contains the properties that are unique to linked OLE objects. Read-only.

lock aspect ratio

> **True** if the specified shape retains its original proportions when you resize it. **False** if you can change the height and width of the shape independently of one another when you resize it. Read/write.

media type

> Returns the OLE media type. Read-only.
>
> Can be one of the following:

- **media type unset**
- **media type movie**
- **media type other**
- **media type sound**

name

> Returns or sets the shape name. Read/write.

rotation

> Returns or sets the number of degrees the specified shape is rotated around the z-axis. A positive value indicates clockwise rotation; a negative value indicates counterclockwise rotation. Read/write.
>
> To set the rotation of a three-dimensional shape around the x-axis or the y-axis, use the *x rotation* property or the *y rotation* property of the **threeD format** object.

shadow format

Returns a **shadow format** object that represents the shadow formatting for the specified shape. Read-only.

shape type

Returns the shape type. Read-only.

Can be one of the following:

shape type auto	**shape type linked OLE object**
shape type callout	**shape type linked picture**
shape type chart	**shape type media**
shape type comment	**shape type embedded OLE control**
shape type embedded OLE object	**shape type picture**
shape type form control	**shape type placeholder**
shape type free form	**shape type unset**
shape type group	**shape type text box**
shape type line	**shape type text effect**

text frame

Returns a **text frame** object that contains the text for the specified shape. Read-only.

threeD format

Returns a **threeD format** object that contains 3-D-effect formatting properties for the specified shape. Read-only.

top

Returns or sets the vertical position (in points) of the specified shape or shape range. Read/write.

vertical flip

True if the specified shape is flipped around the vertical axis. Read-only.

visible

True if the specified object, or the formatting applied to it, is visible. Read/write.

width

Returns or sets the width (in points) of the specified object. Read/write.

z order position

Returns the position of the object in the z-order, which corresponds to the object's index number. Read-only.

Whenever you make a new shape or object, it's added to the front of the z-order by default.

Class: shape table

Plural

shape tables

Represents a table shape on a slide. The **shape table** class inherits all the properties of the **shape** class.

Use **shape table** *index*, where *index* is the name or the index number, to return a single **shape table** object.

This example creates a new table on slide two of the active presentation. The table has three rows and four columns. It is 10 points from the left edge of the slide, and 10 points from the top edge. The width of the new table is 288 points, which makes each of the four columns one inch wide (there are 72 points per inch). The height is set to 216 points, which makes each of the three rows one inch tall.

```
set myPres to slide 2 of active presentation
set shpTable to make new shape table at the beginning of myPres ¬
        with properties {number of rows:3, number of columns:4}
set left position of shpTable to 10
set top of shpTable to 10
set width of shpTable to 288
set height of shpTable to 216
```

Properties

number of columns

> Returns the number of columns in the shape table. Read-only.

number of rows

> Returns the number of rows in the shape table. Read-only

table object

> Returns a **table** object that represents a table in a shape. Read-only.

Class: text box

Plural

text boxes

Represents a text box object in the drawing layer of a document. The **text box** class inherits all the properties of the **shape** class.

Use **text box** *index*, where *index* is the name or the index number, to return a single **text box** object.

This example adds a text box that contains the text "Test Box" to `myPres`.

```
set myPres to slide 1 of active presentation
set shpTxtBx to make new text box at the beginning of myPres with properties ¬
        {text orientation:horizontal, left position:100, top:100, ¬
        width:300, height:200}
set content of text range of text frame of shpTxtBx to "Test Box"
```

Properties

<*Inheritance*> **shape**

Inherits the properties and elements of the **shape** class.

text orientation

Returns the orientation of the text inside the text box.

Can be one of the following:

- **downward**

- **horizontal**

- **text orientation unset**

- **vertical east asian**

- **vertical**

- **horizontal rotated east asian**

- **upward**

Class: text frame

Represents the text frame in a **shape** object. Contains the text in the text frame as well as the properties that control the alignment and anchoring of the text frame.

Use the *text frame* property to return a **text frame** object. The following example adds a rectangle to myPres, adds text to the rectangle, and then sets the margins for the text frame.

```
set myPres to slide 1 of active presentation
set shpRect to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape rectangle, left position:0, top:0, ¬
        width:250, height:140}
set tfShpRect to text frame of shpRect
set content of text range of tfShpRect to "Here is some test text"
set bottom margin of tfShpRect to 10
set left margin of tfShpRect to 10
set right margin of tfShpRect to 10
set top margin of tfShpRect to 10
```

Use the *has text frame* property to determine whether a shape has a text frame, and use the *has text* property to determine whether the text frame contains text, as shown in the following example.

```
set myPres to slide 1 of active presentation
repeat with s in (get shapes of myPres)
    if has text frame of s is true then
        if has text of text frame of s is true then
         display dialog (get content of text range of text frame of s)
        end if
    end if
end repeat
```

Properties

auto size

> **True** if the size of the specified object is changed automatically to fit text within its boundaries. Read/write.

bottom margin

> Returns or sets the distance (in points) between the bottom of the text frame and the bottom of the inscribed rectangle of the shape that contains the text. Read/write.

> This property is ignored when the *auto margins* property is set to **true**.

has text

> **True** if the specified text frame has text associated with it. Read-only.

horizontal anchor

> Returns or sets the horizontal anchor type for the specified text. Read/write.

> Can be one of the following:

> - **center**
> - **none**
> - **horizontal anchor unset**

left margin

> Returns or sets the distance (in points) between the left edge of the text frame and the left edge of the inscribed rectangle of the shape that contains the text. Read/write.

> This property is ignored when the *auto margins* property is set to **true**.

right margin

Returns or sets the distance (in points) between the right edge of the text frame and the right edge of the inscribed rectangle of the shape that contains the text. Read/write.

This property is ignored when the *auto margins* property is set to **true**.

ruler

Returns a **ruler** object that represents the ruler for the specified text. Read-only.

text orientation

Returns or sets the orientation of the text inside the frame. Read/write.

Can be one of the following:

- **downward**
- **horizontal**
- **text orientation unset**
- **upward**
- **vertical east asian**
- **vertical**
- **horizontal rotated east asian**

text range

Returns a **text range** object that represents the text in the specified text frame. Read-only.

top margin

Returns or sets the distance (in points) between the top of the text frame and the top of the inscribed rectangle of the shape that contains the text. Read/write.

This property is ignored when the *auto margins* property is set to **true**.

vertical anchor

Returns or sets the vertical anchor type for the specified text. Read/write.

Can be one of the following:

- **anchor bottom**
- **anchor bottom baseline**
- **anchor middle**
- **anchor top**
- **anchor top baseline**
- **vertical anchor unset**

word wrap

True if lines break automatically to fit inside the shape. Read/write.

Class: threeD format

Represents a shape's three-dimensional formatting.

Use the *threeD format* property to return a **threeD format** object. The following example adds an oval to `myPres` and then specifies that the oval be extruded to a depth of 50 points and that the extrusion be purple.

```
set myPres to slide 1 of active presentation

set shpOval to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape oval, left position:90, top:90, ¬
        width:90, height:40}

set visible of threeD format of shpOval to true

set depth of threeD format of shpOval to 50

set extrusion color of threeD format of shpOval to ¬
        ({255, 100, 255} as RGB color)
```

Remarks

You cannot apply three-dimensional formatting to some kinds of shapes, such as beveled shapes or multiple-disjoint paths. Most of the properties of the **threeD format** object for such a shape will fail.

Properties

depth

> Returns or sets the depth of the shape's extrusion. Can be a value from -600 through 9600 (positive values produce an extrusion whose front face is the original shape; negative values produce an extrusion whose back face is the original shape). Read/write.

extrusion color

> Returns or sets an RGB color that represents the color of the shape's extrusion. Read/write.

format

> Returns the preset extrusion format. Each preset extrusion format contains a set of preset values for the various properties of the extrusion. If the extrusion has a custom format rather than a preset format, this property returns **preset threeD format unset**. Read-only.

> Can be one of the following:

> | | | |
> |---|---|---|
> | **preset threeD format unset** | **format7** | **format14** |
> | **format1** | **format8** | **format15** |
> | **format2** | **format9** | **format16** |
> | **format3** | **format10** | **format17** |
> | **format4** | **format11** | **format18** |
> | **format5** | **format12** | **format19** |
> | **format6** | **format13** | **format20** |

> The values for this property correspond to the options (numbered from left to right, top to bottom) displayed when you click the **3-D** button on the **Drawing** toolbar.

> To set the preset extrusion format, use the **set threeD format** command.

perspective

> **True** if the extrusion appears in perspective — that is, if the walls of the extrusion narrow toward a vanishing point. **False** if the extrusion is a parallel, or orthographic, projection — that is, if the walls don't narrow toward a vanishing point. Read/write.

preset extrusion direction

> Returns the direction that the extrusion's sweep path takes away from the extruded shape (the front face of the extrusion). Read-only.

> Can be one of the following:

> - **extrude bottom**
> - **extrude bottom left**
> - **extrude bottom right**
> - **extrude left**
> - **extrude none**
> - **extrude right**
> - **extrude top**
> - **extrude top left**
> - **extrude top right**
> - **preset extrusion direction unset**

> To set the value of this property, use the **set extrusion direction** command.

preset lighting direction

> Returns or sets the position of the light source relative to the extrusion. Read/write.

> Can be one of the following:

> - **light from bottom**
> - **light from bottom left**
> - **light from bottom right**
> - **light from left**
> - **light from none**
> - **light from right**
> - **light from top**
> - **light from top left**
> - **light from top right**
> - **preset lighting direction unset**

> **Note** You won't see the lighting effects you set if the extrusion has a wireframe surface.

preset lighting softness

> Returns or sets the intensity of the extrusion lighting. Read/write.
>
> Can be one of the following:

- **lighting bright**
- **lighting dim**
- **lighting normal**
- **lighting softness unset**

preset material

> Returns or sets the extrusion surface material. Read/write.
>
> Can be one of the following:

- **matte**
- **metal**
- **plastic**
- **wireframe**
- **preset material unset**

visible

> **True** if the specified object, or the formatting applied to it, is visible. Read/write.

x rotation

> Returns or sets the rotation (in degrees) of the extruded shape around the x-axis. Can be a value from -90 through 90. A positive value indicates upward rotation; a negative value indicates downward rotation. Read/write.
>
> To set the rotation of the extruded shape around the y-axis, use the *rotation y* property. To set the rotation of the extruded shape around the z-axis, use the *rotation* property of the **shape** object. To change the direction of the extrusion's sweep path without rotating the front face of the extrusion, use the **set extrusion direction** command.

y rotation

> Returns or sets the rotation (in degrees) of the extruded shape around the y-axis. Can be a value from -90 through 90. A positive value indicates rotation to the left; a negative value indicates rotation to the right. Read/write.
>
> To set the rotation of the extruded shape around the x-axis, use the *rotation x* property. To set the rotation of the extruded shape around the z-axis, use the *rotation* property of the **shape** object. To change the direction of the extrusion's sweep path without rotating the front face of the extrusion, use the **set extrusion direction** command.

Class: word art

Plural

word arts

Represents a word art object.

Use **word art** *index*, where *index* is the name or the index number, to return a single **word art** object. This example adds WordArt that contains the text "Test Text" at the beginning of the first slide of the active presentation.

```
set myPres to slide 1 of active presentation
make new word art at beginning of myPres with properties ¬
        {word art text:"Test text", left position:72, top:72}
set preset word art effect of word art format of word art 1 of myPres ¬
        to text effect11
set font name of word art format of word art 1 of myPres ¬
        to "Arial Black"
set font size of word art format of word art 1 of myPres to 36
set bold of word art format of word art 1 of myPres to true
```

Properties

<Inheritance>   **shape**

Inherits the properties and elements of the **shape** class.

bold

True to set the font used in the WordArt to bold. Read-only.

font italic

True to set the font used in the WordArt to italic. Read-only.

font name

The name of the font used in the WordArt. Read-only.

font size

The size (in points) of the font used in the WordArt. Read-only.

preset word art effect

> Returns the style of the specified WordArt. The values for this property correspond to the formats in the **WordArt Gallery** dialog box (numbered from left to right, top to bottom). Read-only.

> Can be one of the following.:

text effect unset	**text effect11**	**text effect21**
text effect1	**text effect12**	**text effect22**
text effect2	**text effect13**	**text effect23**
text effect3	**text effect14**	**text effect24**
text effect4	**text effect15**	**text effect25**
text effect5	**text effect16**	**text effect26**
text effect6	**text effect17**	**text effect27**
text effect7	**text effect18**	**text effect28**
text effect8	**text effect19**	**text effect29**
text effect9	**text effect20**	**text effect30**
text effect10		

word art format

> Returns the **word art format** object associated with the WordArt shape object. Read-only.

word art text

> The text in the WordArt. Read-only

Class: word art format

Contains properties that apply to WordArt objects.

Use the *word art format* property to return a **word art format** object. The following example sets the font name and formatting for shape one on `myDocument`. For this example to work, shape one must be a WordArt object.

```
set myPres to slide 1 of active presentation

set theWAF to word art format of word art 1 of myPres

set font name of theWAF to "Courier New"

set font bold of theWAF to true

set font italic of theWAF to true
```

Properties

font bold

> **True** if the text of the WordArt shape is formatted as bold. Read/write.

font italic

> **True** if the text of the WordArt shape is formatted as italic. Read/write.

font name

> Returns or sets the name of the font used by the WordArt shape. Read/write.

kerned pairs

> **True** if character pairs in the specified WordArt are kerned. Read/write.

normalized height

> **True** if all characters (both uppercase and lowercase) in the specified WordArt are the same height. Read/write.

preset shape

> Returns or sets the shape of the specified WordArt. Read/write.

> Can be one of the following:

arch down curve	**double wave1**
arch down pour	**double wave2**
arch up curve	**fade down**
arch up pour	**fade left**
button curve	**fade right**
button pour	**fade up**
can down	**inflate**
can up	**inflate bottom**
cascade down	**inflate top**
cascade up	**text effect unset**
chevron down	**plain text**
chevron up	**ring inside**
circle curve	**ring outside**
circle pour	**slant down**
curve down	**slant up**
curve up	**stop**
deflate	**triangle down**
deflate bottom	**triangle up**
deflate inflate	**wave1**
deflate inflate deflate	**wave2**
deflate top	

> Setting the *preset word art* property automatically sets the *preset shape* property.

preset word art

Returns or sets the style of the specified WordArt. The values for this property correspond to the formats in the **WordArt Gallery** dialog box (numbered from left to right, top to bottom). Read/write.

Can be one of the following:

text effect unset	**text effect16**
text effect1	**text effect17**
text effect2	**text effect18**
text effect3	**text effect19**
text effect4	**text effect20**
text effect5	**text effect21**
text effect6	**text effect22**
text effect7	**text effect23**
text effect8	**text effect24**
text effect9	**text effect25**
text effect10	**text effect26**
text effect11	**text effect27**
text effect12	**text effect28**
text effect13	**text effect29**
text effect14	**text effect30**
text effect15	

Setting the *preset word art effect* property automatically sets many other formatting properties of the specified shape.

rotated chars

True if characters in the specified WordArt are rotated 90 degrees relative to the WordArt's bounding shape. **False** if characters in the specified WordArt retain their original orientation relative to the bounding shape. Read/write.

If the WordArt has horizontal text, setting the *rotated chars* property to **true** rotates the characters 90 degrees counterclockwise. If the WordArt has vertical text, setting the *rotated chars* property to **false** rotates the characters 90 degrees clockwise. To switch between horizontal and vertical text flow, use the **toggle vertical text** command.

The **flip** command and *rotation* property of the **shape** object and the *rotated chars* property and **toggle vertical text** command all affect the character orientation and direction of text flow in a **word art** object. You may have to experiment to find out how to combine the effects of these properties and commands to get the result you want.

text alignment

Returns or sets the alignment for the specified text effect. Read/write.

Can be one of the following:

- **text effect alignment unset**
- **left text effect alignment**
- **centered text effect alignment**
- **right text effect alignment**
- **justify text effect alignment**
- **word justify text effect alignment**
- **stretch justify text effect alignment**

tracking

Returns or sets the ratio of the horizontal space allotted to each character in the specified WordArt to the width of the character. Can be a value from 0 (zero) through 5. (Large values for this property specify ample space between characters; values less than 1 can produce character overlap.) Read/write.

The following table gives the values of the *tracking* property that correspond to the settings available in the user interface.

User interface setting	Equivalent tracking property value
Very Tight	0.8
Tight	0.9
Normal	1.0
Loose	1.2
Very Loose	1.5

word art text

Returns or sets the text associated with the WordArt. Read/write..

Drawing Suite Commands

Command: `apply`

Applies to the specified shape formatting that's been copied by using the **pick up** command.

Syntax

apply   shape   Required. An expression that returns a **shape**.

Example

This example copies the formatting of shape one on `mySlide` and then applies the copied formatting to shape two.

```
set mySlide to slide 1 of active presentation

pick up shape 1 of mySlide

apply shape 2 of mySlide
```

Command: `automatic length`

Specifies that the first segment of the callout line (the segment attached to the text callout box) be scaled automatically when the callout is moved. Use the **custom length** command to specify that the first segment of the callout line retain the fixed length returned by the *callout format length* property whenever the callout is moved. Applies only to callouts whose lines consist of more than one segment (types **callout three** and **callout four**).

Syntax

automatic length callout/callout format Required. An expression that returns a **callout** or **callout format** object.

Remarks

Applying this command sets the *auto length* property to **true**.

Example

This example toggles between an automatically scaling first segment and one with a fixed length for the callout line for shape one on `myDocument`. For the example to work, shape one must be a callout.

```
set mySlide to slide 1 of active presentation
if auto length of callout format of callout 1 of mySlide is true then
    custom length callout format of callout 1 of mySlide length 50
else
    automatic length callout format of callout 1 of mySlide
end if
```

Command: `begin connect`

Attaches the beginning of the specified connector to a specified shape. If there's already a connection between the beginning of the connector and another shape, that connection is broken. If the beginning of the connector isn't already positioned at the specified connecting site, this command moves the beginning of the connector to the connecting site and adjusts the size and position of the connector. To attach the end of the connector to a shape, use the **end connect** command.

Syntax

begin connect connector/connector format Required. An expression that returns a **connector** or **connector format** object.

  connected shape shape Required. The shape to attach the beginning of the connector to.

  connection site integer Required. A connection site on the shape specified by **connected shape**. Must be an integer between 1 and the integer returned by the *connection site count* property of the specified shape. If you want the connector to automatically find the shortest path between the two shapes it connects, specify any valid integer for this argument and then use the **reroute connections** command after the connector is attached to shapes at both ends.

Remarks

When you attach a connector to an object, the size and position of the connector are automatically adjusted, if necessary.

Example

This example adds two shapes to `myPres` and connects them with a curved connector. Note that when you attach the connector to the shapes, the size and position of the connector are automatically adjusted; therefore, the position and dimensions you specify when adding the connector are irrelevant (dimensions must be nonzero).

```
set myPres to slide 1 of active presentation
set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}
set shpTri to make new shape at the end of myPres with properties ¬
        {auto Shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}
set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}
begin connect shpCon connected shape shpTrap connection site 1
end connect shpCon connected shape shpTri connection site 1
reroute connections shpCon
```

Command: `begin disconnect`

Detaches the beginning of the specified connector from the shape it's attached to. This command doesn't alter the size or position of the connector: the beginning of the connector remains positioned at a connection site but is no longer connected. To detach the end of the connector from a shape, use the **end disconnect** command.

Syntax

begin disconnect connector/connector format   Required. An expression that returns a **connector format** object.

Example

This example adds two shapes to `myPres`, attaches them with a connector, automatically reroutes the connector along the shortest path, and then detaches the connector from the shapes.

```
set myPres to slide 1 of active presentation
set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}
set shpTri to make new shape at the end of myPres with properties ¬
        {auto Shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}
set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}
begin connect shpCon connected shape shpTrap connection site 1
end connect shpCon connected shape shpTri connection site 1
reroute connections shpCon
begin disconnect shpCon
end disconnect shpCon
```

Command: copy shape

Copies the specified shape to the Clipboard.

Syntax

copy shape   shape   Required. An expression that returns **shape** object.

Example

This example copies shape one on slide one in the active presentation to the Clipboard and then pastes it onto slide two.

```
copy shape (shape 1 of slide 1 of active presentation)
go to slide view of active window number 1
paste object view of active window
```

Command: custom drop

Sets the vertical distance (in points) from the edge of the text bounding box to the place where the callout line attaches to the text box. This distance is measured from the top of the text box unless the *auto attach* property is set to **true** and the text box is to the left of the origin of the callout line (the place that the callout points to), in which case the drop distance is measured from the bottom of the text box.

Syntax

custom drop   callout/callout format   Required. An expression that returns a **callout format** object.

  drop amount   small real   Required. The drop distance, in points.

Example

This example sets the custom drop distance to 14 points, and specifies that the drop distance always be measured from the top.

```
set myPres to slide 1 of active presentation

custom drop callout 3 of myPres drop amount 14

set auto attach of callout format of callout 3 of myPres to false
```

Command: custom length

Specifies that the first segment of the callout line (the segment attached to the text callout box) retain a fixed length whenever the callout is moved. To specify that the first segment of the callout line be scaled automatically whenever the callout is moved, use the **automatic length** command. Applies only to callouts whose lines consist of more than one segment (types **callout three** and **callout four**).

Syntax

custom length callout/callout format Required. An expression that returns a **callout** or **callout format** object.

   length small real Required. The length (in points) of the first segment of the callout.

Remarks

Applying this command sets the *auto length* property to **false** and sets the *callout format length* property to the value specified for the **length** argument.

Example

This example toggles between an automatically scaling first segment and one with a fixed length for the callout line for shape one on `myPres`. For the example to work, shape one must be a callout.

```
set myPres to slide 1 of active presentation

if auto length of callout format of callout 1 of myPres is true then

    custom length callout format of callout 1 of myPres length 50

else

    automatic length callout format of callout 1 of myPres

end if
```

Command: cut shape

Deletes the specified object and places it on the Clipboard.

Syntax

cut shape shape Required. An expression that returns a **shape** object.

Example

This example deletes shape one from slide one in the active presentation, places a copy of it on the Clipboard, and then pastes it onto slide two.

```
cut shape (shape 1 of slide 1 of active presentation)

go to slide view of active window number 2

paste object view of active window
```

Command: end connect

Attaches the end of the specified connector to a specified shape. If there's already a connection between the end of the connector and another shape, that connection is broken. If the end of the connector isn't already positioned at the specified connecting site, this command moves the end of the connector to the connecting site and adjusts the size and position of the connector. To attach the beginning of the connector to a shape, use the **begin connect** command.

Syntax

end connect connector/connector format   Required. An expression that returns a **connector** or **connector format** object.

> **connected shape** shape   Required. The shape to attach the end of the connector to.

> **connection site** integer   Required. A connection site on the shape specified by **connected shape**. Must be an integer between 1 and the integer returned by the *connection site count* property of the specified shape. If you want the connector to automatically find the shortest path between the two shapes it connects, specify any valid integer for this argument and then use the **reroute connections** command after the connector is attached to shapes at both ends.

Remarks

When you attach a connector to an object, the size and position of the connector are automatically adjusted, if necessary.

Example

This example adds two shapes to `myPres` and connects them with a curved connector. Note that when you attach the connector to the shapes, the size and position of the connector are automatically adjusted; therefore, the position and dimensions you specify when adding the connector are irrelevant (dimensions must be nonzero).

```
set myPres to slide 1 of active presentation
set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}
set shpTri to make new shape at the end of myPres with properties ¬
        {auto shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}
set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}
begin connect shpCon connected shape shpTrap connection site 1
end connect shpCon connected shape shpTri connection site 1
reroute connections shpCon
```

Command: end disconnect

Detaches the end of the specified connector from the shape it's attached to. This command doesn't alter the size or position of the connector: the end of the connector remains positioned at a connection site but is no longer connected. To detach the beginning of the connector from a shape, use the **begin disconnect** command.

Syntax

end disconnect connector/connector format Required. An expression that returns a **connector** or **connector format** object.

Example

This example adds two shapes to myPres, attaches them with a connector, automatically reroutes the connector along the shortest path, and then detaches the connector from the shapes.

```
set myPres to slide 1 of active presentation
set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}
set shpTri to make new shape at the end of myPres with properties ¬
        {auto shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}
set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}
begin connect shpCon connected shape shpTrap connection site 1
end connect shpCon connected shape shpTri connection site 1
reroute connections shpCon
begin disconnect shpCon
end disconnect shpCon
```

Command: flip

Flips the specified shape around its horizontal or vertical axis.

Syntax

flip shape Required. An expression that returns a **shape** object.

   direction enumeration Required. Specifies whether the shape is to be flipped horizontally or vertically. Can be either of the following: **flip horizontal** or **flip vertical**.

Example

This example adds a triangle to `mySlide`, duplicates the triangle, and then flips the duplicate triangle vertically and makes it red.

```
set mySlide to slide 1 of active presentation
set shpTri to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape right triangle, left position:10, top:10, ¬
        width:50, height:50}
set fore color of fill of shpTri to ({255, 0, 0} as RGB color)
flip shpTri direction flip vertical
```

Command: get action setting for

Returns an **action setting** object that contains information about what action occurs when the user clicks or moves the mouse over the specified shape or text range during a slide show.

Syntax

get action setting for   shape   An expression that returns a **shape** object.

  event   enumeration   Required. The mouse action that triggers the action setting. Can be one of the following: **mouse activation mouse click** or **mouse activation mouse over**.

Example

The following example sets the actions for clicking and moving the mouse over shape one on slide two in the active presentation.

```
set myShape to shape 1 of slide 2 of active presentation
set mClick to get action setting for myShape ¬
        event mouse activation mouse click
set action of mClick to action type last slide
set mOver to get action setting for myShape ¬
        event mouse activation mouse over
set name of action sound effect of mOver to "applause.wav"
```

Command: one color gradient

Sets the specified fill to a one-color gradient.

Syntax

one color gradient   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

  style   enumeration   Required. The gradient style. Can be one of the following: **diagonal down gradient**, **diagonal up gradient**, **from center gradient**, **from corner gradient**, **gradient unset**, **from title gradient**, **horizontal gradient**, or **vertical gradient**.

  variant   integer   Required. The gradient variant. Can be a value from 1 to 4, corresponding to the four variants on the **Gradient** tab in the **Fill Effects** dialog box. If **style** is **gradient from title** or **gradient from center**, this argument can be either 1 or 2.

  degree   small real   Required. The gradient degree. Can be a value from 0.0 (dark) to 1.0 (light).

Example

This example adds a rectangle with a one-color gradient fill to mySlide.

```
set mySlide to slide 1 of active presentation
set shpRect to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:90, top:90, ¬
        width:90, height:80}
set fore color of fill of shpRect to ({0, 128, 128} as RGB color)
one color gradient shpRect style horizontal gradient variant 1 degree 1
```

Command: `patterned`

Sets the specified fill to a pattern.

Syntax

patterned   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

   pattern   enumeration Required. The pattern to be used for the specified fill. Can be one of the following:

pattern unset	**wide downward diagonal**
five percent pattern	**wide upward diagonal pattern**
ten percent pattern	**dashed upward diagonal pattern**
twenty percent pattern	**dashed downward diagonal pattern**
twenty five percent pattern	**narrow vertical pattern**
thirty percent pattern	**narrow horizontal pattern**
forty percent pattern	**dashed vertical pattern**
fifty percent pattern	**dashed horizontal pattern**
sixty percent pattern	**large confetti pattern**
seventy percent pattern	**large grid pattern**
seventy five percent pattern	**horizontal brick pattern**
eighty percent pattern	**large checker board pattern**
ninety percent pattern	**small confetti pattern**
dark horizontal pattern	**zig zag pattern**
dark vertical pattern	**solid diamond pattern**
dark downward diagonal pattern	**diagonal brick pattern**
dark upward diagonal pattern	**outlined diamond pattern**
small checker board pattern	**plaid pattern**
trellis pattern	**sphere pattern**
light horizontal pattern	**weave pattern**
light vertical pattern	**dotted grid pattern**
light downward diagonal pattern	**divot pattern**
light upward diagonal pattern	**shingle pattern**
small grid pattern	**wave pattern**
dotted diamond pattern	

Remarks

To set the colors used in the pattern, use the *back color* and *fore color* properties.

Example

This example adds an oval with a patterned fill to `mySlide`.

```
set mySlide to slide 1 of active presentation
set shpOval to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape oval, left position:60, top:60, ¬
        width:80, height:40}
set fore color of fill of shpOval to ({128, 0, 0} as RGB color)
set back color of fill of shpOval to ({0, 0, 255} as RGB color)
patterned fill of shpOval pattern dark vertical pattern
```

Command: `pick up`

Copies the formatting of the specified shape. To apply the copied formatting to another shape, use the **apply** command.

Syntax

pick up shape   Required. An expression that returns a **shape**.

Example

This example copies the formatting of shape one on `mySlide` and then applies the copied formatting to shape two.

```
set mySlide to slide 1 of active presentation
pick up shape 1 of mySlide
apply shape 2 of mySlide
```

Command: `preset drop`

Specifies whether the callout line attaches to the top, bottom, or center of the callout text box or whether it attaches at a point that's a specified distance from the top or bottom of the text box.

Syntax

preset drop callout format   Required. An expression that returns a **callout format** object.

  drop type enumeration   Required. The starting position of the callout line relative to the text bounding box. Can be one of the following: **drop bottom**, **drop center**, **drop unset**, or **drop top**. Specifying **drop custom** for this argument will cause your code to fail.

Example

This example specifies that the callout line attach to the top of the text bounding box for shape one on `mySlide`.

```
set mySlide to slide 1 of active presentation
preset drop callout format of callout 1 of mySlide drop type drop top
```

This example toggles between two preset drops for shape one on `mySlide`.

```
set mySlide to slide 1 of active presentation
if drop type of callout format of callout 1 of mySlide is drop top then
    preset drop callout format of callout 1 of mySlide drop type drop bottom
else
    preset drop callout format of callout 1 of mySlide drop type drop top
end if
```

Command: `preset gradient`

Sets the specified fill to a preset gradient.

Syntax

preset gradient   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

   style   enumeration   Required. The gradient style. Can be one of the following: **diagonal down gradient**, **diagonal up gradient**, **from center gradient**, **from corner gradient**, **gradient unset**, **from title gradient**, **horizontal gradient**, or **vertical gradient**.

   variant   integer   Required. The gradient variant. Can be a value from 1 to 4, corresponding to the four variants on the **Gradient** tab in the **Fill Effects** dialog box. If **style** is **gradient from title** or **gradient from center**, this argument can be either 1 or 2.

   gradient type   enumeration   Required. The gradient type. Can be one of the following:

preset gradient unset	**gradient wheat**
gradient early sunset	**gradient parchment**
gradient late sunset	**gradient mahogany**
gradient nightfall	**gradient rainbow**
gradient daybreak	**gradient rainbow2**
gradient horizon	**gradient gold**
gradient desert	**gradient gold2**
gradient ocean	**gradient brass**
gradient calm water	**gradient chrome**
gradient fire	**gradient chrome2**
gradient fog	**gradient silver**
gradient moss	**gradient sapphire**
gradient peacock	

Example

This example adds a rectangle with a preset gradient fill to `mySlide`.

```
set mySlide to slide 1 of active presentation
set shpRect to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:90, top:90, ¬
        width:140, height:80}
preset gradient fill of shpRect style horizontal gradient variant 1 ¬
        gradient type gradient brass
```

Command: `preset textured`

Sets the specified fill to a preset texture.

Syntax

preset textured   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

   texture   enumeration   Required. The preset texture. Can be one of the following:

preset texture unset	**texture newsprint**
texture papyrus	**texture recycled paper**
texture canvas	**texture parchment**
texture denim	**texture stationery**
texture woven mat	**texture blue tissue paper**
texture water droplets	**texture pink tissue paper**
texture paper bag	**texture purple mesh**
texture fish fossil	**texture bouquet**
texture sand	**texture cork**
texture green marble	**texture walnut**
texture white marble	**texture oak**
texture brown marble	**texture medium wood**
texture granite	

Example

This example adds a rectangle with a green-marble textured fill to `mySlide`.

```
set mySlide to slide 1 of active presentation
set shpCan to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape can, left position:90, top:90, ¬
        width:40, height:80}
preset textured fill of shpCan texture texture green marble
```

Command: `reroute connections`

Reroutes connectors so that they take the shortest possible path between the shapes they connect. To do this, the **reroute connections** command may detach the ends of a connector and reattach them to different connecting sites on the connected shapes.

This command reroutes all connectors attached to the specified shape; if the specified shape is a connector, it's rerouted.

Syntax

reroute connections   shape   Required. An expression that returns a **shape** object.

Remarks

If this command is applied to a connector, only that connector will be rerouted. If this command is applied to a connected shape, all connectors to that shape will be rerouted.

Example

This example adds two shapes to `myPres` and connects them with a curved connector. Note that when you attach the connector to the shapes, the size and position of the connector are automatically adjusted; therefore, the position and dimensions you specify when adding the connector are irrelevant (dimensions must be nonzero).

```
set myPres to slide 1 of active presentation
set shpTrap to make new shape at the beginning of myPres with properties ¬
        {auto shape type:autoshape trapezoid,left position:100, top:50, ¬
        width:200, height:100}
set shpTri to make new shape at the end of myPres with properties ¬
        {auto Shape type:autoshape right triangle, left position:300, ¬
        top:300, width:200, height:100}
set shpCon to make new connector at the end of myPres with properties ¬
        {connector type:curve, left position:0, top:0, width:100, height:100}
begin connect shpCon connected shape shpTrap connection site 1
end connect shpCon connected shape shpTri connection site 1
reroute connections shpCon
```

Command: reset rotation

Resets the extrusion rotation around the x-axis and the y-axis to 0 (zero) so that the front of the extrusion faces forward. This command doesn't reset the rotation around the z-axis.

Syntax

reset rotation   shape/threeD format   Required. An expression that returns a **shape** or **threeD format** object.

Remarks

To set the extrusion rotation around the x-axis and the y-axis to anything other than 0 (zero), use the *x rotation* and *y rotation* properties of the **threeD format** object. To set the extrusion rotation around the z-axis, use the *rotation* property of the **shape** object that represents the extruded shape.

Example

This example resets the rotation around the x-axis and the y-axis to 0 (zero) for the extrusion of shape one on `mySlide`.

```
set mySlide to slide 1 of active presentation
reset rotation threeD format of shape 1 of mySlide
```

Command: `scale height`

Scales the height of the picture by a specified factor. You can indicate whether you want to scale the shape relative to the original size or relative to the current size.

Syntax

scale height picture   Required. An expression that returns a **picture** object.

> **factor** small real   Required. Specifies the ratio between the height of the shape after you resize it and the current or original height. For example, to make a rectangle 50 percent larger, specify 1.5 for this argument.
>
> **relative to original size** Boolean   Required. **True** to scale the shape relative to its original size. **False** to scale it relative to its current size.
>
> **scale** enumeration   Required. The part of the shape that retains its position when the shape is scaled. Can be one of the following: **scale from bottom right**, **scale from middle**, or **scale from top left**.

Example

This example scales all pictures on `mySlide` to 175 percent of their original height and width, and it scales all other shapes to 175 percent of their current height and width.

```
set mySlide to slide 1 of active presentation
repeat with s in (get shapes of mySlide)
    if shape type of s is shape type picture then
        scale height s factor 1.75 scale scale from top left ¬
            with relative to original size
        scale width s factor 1.75 scale scale from top left ¬
            with relative to original size
    else
        scale height s factor 1.75 scale scale from top left ¬
            without relative to original size
        scale width s factor 1.75 scale scale from top left ¬
            without relative to original size
    end if
end repeat
```

Command: `scale width`

Scales the width of the shape by a specified factor. You can indicate whether you want to scale the shape relative to the original size or relative to the current size.

Syntax

scale width picture   Required. An expression that returns a **shape** object.

> **factor** small real   Required. Specifies the ratio between the width of the shape after you resize it and the current or original width. For example, to make a rectangle 50 percent larger, specify 1.5 for this argument.

> **relative to original size** Boolean   Required. **True** to scale the shape relative to its original size. **False** to scale it relative to its current size.

> **scale** enumeration   Optional. The part of the shape that retains its position when the shape is scaled. Can be one of the following: **scale from bottom right**, **scale from middle**, or **scale from top left**.

Example

This example scales all pictures on `mySlide` to 175 percent of their original height and width, and it scales all other shapes to 175 percent of their current height and width.

```
set mySlide to slide 1 of active presentation
repeat with s in (get shapes of mySlide)
    if shape type of s is shape type picture then
        scale height s factor 1.75 scale scale from top left ¬
            with relative to original size
        scale width s factor 1.75 scale scale from top left ¬
            with relative to original size
    else
        scale height s factor 1.75 scale scale from top left ¬
            without relative to original size
        scale width s factor 1.75 scale scale from top left ¬
            without relative to original size
    end if
end repeat
```

Command: `set shapes default properties`

Applies the formatting for the specified shape to the default shape. Shapes created after this command has been used will have this formatting applied to them by default.

Syntax

set shapes default properties   shape   Required. An expression that returns a **shape** object.

Example

This example adds a rectangle to `mySlide`, formats the rectangle's fill, applies the rectangle's formatting to the default shape, and then adds another smaller rectangle to the document. The second rectangle has the same fill as the first one.

```
set mySlide to slide 1 of active presentation
set shpRect to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:5, top:5, ¬
        width:80, height:60}
set fore color of fill of shpRect to ({0, 0, 255} as RGB color)
set back color of fill of shpRect to ({0, 204, 255} as RGB color)
patterned shpRect pattern horizontal brick pattern
set shapes default properties shpRect
set newShp to make new shape at the end of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:90, top:90, ¬
        width:40, height:30}
```

Command: `solid`

Sets the specified fill to a uniform color. Use this command to convert a gradient, textured, patterned, or background fill back to a solid fill.

Syntax

solid   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

Example

This example converts all fills on `mySlide` to uniform red fills.

```
set mySlide to slide 1 of active presentation
repeat with s in (get shapes of mySlide)
    solid s
    set fore color of fill of s to ({255, 0, 0} as RGB color)
end repeat
```

Command: `toggle vertical text`

Switches the text flow in the specified WordArt from horizontal to vertical, or vice versa.

Syntax

toggle vertical text   word art format   Required. An expression that returns a **word art format** object.

Remarks

Using the **toggle vertical text** command swaps the values of the *width* and *height* properties of the **word art** object and leaves the *left position* and *top* properties unchanged.

The **flip** command and *rotation* property of the **shape** object and the **toggle vertical text** command and *rotated chars* property of the **word art format** object all affect the character orientation and the direction of text flow in a **word art** object. You may have to experiment to find out how to combine the effects of these properties and methods to get the result you want.

Example

This example adds WordArt that contains the text "Test" to `mySlide` and switches from horizontal text flow (the default for the specified WordArt style, **text effect1**) to vertical text flow.

```
set mySlide to slide 1 of active presentation
set newWdArt to make new word art at the beginning of mySlide with properties ¬
        {preset word art effect:text effect1, word art text:"Test", ¬
        font name:"Arial Black", font size:36, font bold:true, ¬
        font italic:false, left position:100, top:100}
toggle vertical text word art format of newWdArt
```

Command: `two color gradient`

Sets the specified fill to a two-color gradient.

Syntax

two color gradient   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

> **style**   enumeration   Required. The gradient style. Can be one of the following: **gradient unset**, **diagonal down gradient**, **diagonal up gradient**, **from center gradient**, **from corner gradient**, **from title gradient**, **horizontal gradient**, or **vertical gradient**.

> **variant**   integer   Required. The gradient variant. Can be a value from 1 to 4, corresponding to the four variants on the **Gradient** tab in the **Fill Effects** dialog box. If **style** is **from title gradient** or **from center gradient**, this argument can be either 1 or 2.

Example

This example adds a rectangle with a two-color gradient fill to `myDocument` and sets the background and foreground color for the fill.

```
set mySlide to slide 1 of active presentation
set shpRect to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:0, top:0, ¬
        width:40, height:80}
set fore color of fill of shpRect to ({128, 0, 0} as RGB color)
set back color of fill of shpRect to ({0, 170, 0} as RGB color)
two color gradient fill of shpRect style horizontal gradient variant 1
```

Command: `user picture`

Fills the specified shape with one large image. To fill the shape with small tiles of an image, use the **user textured** command.

Syntax

user picture   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

  picture file   Unicode text   Required. The name of the picture file.

Example

This example adds two rectangles to `mySlide`. The rectangle on the left is filled with one large image of the picture in Tiles.bmp; the rectangle on the right is filled with many small tiles of the picture in Tiles.bmp

```
set mySlide to slide 1 of active presentation
set shpOne to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:0, top:0, ¬
        width:200, height:100}
user picture fill of shpOne picture file "Macintosh HD:Users:Shared:Tiles.bmp"
set shpTwo to make new shape at the end of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:300, top:0, ¬
        width:200, height:100}
user textured fill of shpTwo texture file "Macintosh HD:Users:Shared:Tiles.bmp"
```

Command: `user textured`

Fills the specified shape with small tiles of an image. To fill the shape with one large image, use the **user picture** command.

Syntax

user textured   shape/fill format   Required. An expression that returns a **shape** or **fill format** object.

  texture file   Unicode text   Required. The name of the texture file.

Example

This example adds two rectangles to `mySlide`. The rectangle on the left is filled with one large image of the picture in Tiles.bmp; the rectangle on the right is filled with many small tiles of the picture in Tiles.bmp

```
set mySlide to slide 1 of active presentation
set shpOne to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:0, top:0, ¬
        width:200, height:100}
user picture fill of shpOne picture file "Macintosh HD:Users:Shared:Tiles.bmp"
set shpTwo to make new shape at the end of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:300, top:0, ¬
        width:200, height:100}
user textured fill of shpTwo texture file "Macintosh HD:Users:Shared:Tiles.bmp"
```

Command: z order

Moves the specified shape in front of or behind other shapes in the collection (that is, changes the shape's position in the z-order).

Syntax

z order   shape   Required. An expression that returns a **shape** object.

> **z order position**   enumeration   Required. Specifies where to move the specified shape relative to the other shapes. Can be one of the following: **bring shape forward**, **bring shape to front**, **send shape backward**, or **send shape to back**. The constants **bring shape in front of text** and **send shape behind text** are for use in Microsoft Word only.

Remarks

Use the *z order position* property to determine a shape's current position in the z-order.

Example

This example adds an oval to `mySlide` and then places the oval second from the back in the z-order if there is at least one other shape on the document.

```
set mySlide to slide 1 of active presentation
set myOval to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape oval, left position:100, top:100, ¬
        width:100, height:300}
repeat while z order position of myOval > 2
        z order myOval z order position send shape backward
end repeat
```

Text Suite

Text Suite Classes

Class: character

Plural

characters

A collection of characters in a text range. Each character is an element of a **text range** object.

Use **characters** to return the **characters** in a text range. The following example displays how many characters are in the second shape of slide 2.

```
set theTR to text range of text frame of shape 2 of slide 2 of ¬
        active presentation
display dialog ((count of characters in theTR) as string) & ¬
        " characters"
```

To return a **text range** object that represents one character, use **character** *index*, where *index* is the index number. The index number represents the position of a character in the text range. This example sets the text for shape two on slide one in the active presentation and then makes the second character a subscript character with a 20-percent offset.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
set content of theRange to "H2O"
set theChar to character 2 of theRange
set base line offset of font of theChar to -0.2
```

This example formats every subscript character in shape two on slide one as bold.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
repeat with i from 1 to (get count of characters of theRange)
    if subscript of font of character i of theRange is true then set bold of ¬
        font of character i of theRange to true
end repeat
```

Properties

<*Inheritance*> **text range**

Inherits the properties and elements of the **text range** class.

Class: line

Plural

lines

A collection of lines in a text range. Each line is an element of a **text range** object.

Use **lines** to return the **lines** in a text range. To return a **text range** object that represents one line, use **line** *index*, where *index* is the index number. This example formats as italic the first two lines of shape two on slide one in the active presentation.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
repeat with i from 1 to 2
    set italic of font of line i of theRange to true
end repeat
```

Properties

<*Inheritance*> **text range**

Inherits the properties and elements of the **text range** class.

Class: paragraph

Plural

paragraphs

Represents a single paragraph in a text range. The **paragraphs** list includes all the paragraphs in a text range.

To return a single **paragraph** object, use **paragraph** *index*, where *index* is the index number. The following example right aligns the first paragraph in the specified text range.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
set alignment of paragraph format of paragraph 1 of theRange to ¬
        paragraph align right
```

Properties

<*Inheritance*> **text range**

> Inherits the properties and elements of the **text range** class.

Class: sentence

Plural

sentences

A text range object that represents a sentence in a text range.

Use **sentence** *index*, where *index* is the index number, to return a **text range** object that represents a sentence. The index number represents the position of a sentence in the sentences list. This example formats as bold the second sentence in shape two on slide one in the active presentation.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
set bold of font of sentence 2 of theRange to true
```

Properties

<*Inheritance*> **text range**

> Inherits the properties and elements of the **text range** class.

Class: text flow

Plural

text flows

Represents a run of text in the specified text range. A text run consists of a range of characters that share the same font attributes. There is no text flow object; instead, each text flow is an element of a **text range** object.

To return a **text range** object that represents one text flow, use **text flow** *index*, where *index* is the index number. This example formats the second run in shape two on slide one in the active presentation as bold italic if it's already italic.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
if italic of font of text flow 2 of theRange is true then
    set bold of font of text flow 2 of theRange to true
end if
```

Properties

<*Inheritance*> **text range**

> Inherits the properties and elements of the **text range** class.

Class: text range

Elements

character

word

sentence

line

paragraph

text flow

Contains the text that's attached to a shape, as well as properties and methods for manipulating the text.

To return a **text range** object for any shape you specify, use the *text range* property of the **text frame** object. To return the string of text in the **text range** object, use the *content* property. The following example adds a rectangle to `mySlide` and sets the text it contains.

```
set mySlide to slide 1 of active presentation
set shpRect to make new shape at the beginning of mySlide with properties ¬
        {auto shape type:autoshape rectangle, left position:0, top:0, ¬
        height:250, width:140}
set content of text range of text frame of shpRect to "Here is some test text"
```

To determine whether a shape has a text frame, use the *has text frame* property. To determine whether the **text frame** contains text, use the *has text* property.

Properties

bounds height

Returns the height (in points) of the text bounding box for the specified text frame. Read-only.

bounds width

Returns the width (in points) of the text bounding box for the specified text frame. Read-only.

content

Returns or sets the text contained in the specified text range object. Read/write.

font object

Returns a **font** object that represents character formatting. Read-only.

indent level

Returns or sets the indent level for the specified text as an integer from 1 to 5, where 1 indicates a first-level paragraph with no indentation. Read/write.

left bounds

Returns the distance (in points) from the left edge of the text bounding box for the specified text frame to the left edge of the slide. Read-only.

offset

Returns the position of the first character in the specified text range relative to the first character in the shape that contains the text. Read-only.

paragraph format

Returns a **paragraph format** object that represents paragraph formatting for the specified text. Read-only.

text length

Returns the length (in characters) of the specified text range. Read-only.

top bounds

Returns the distance (in points) from the top of the text bounding box for the specified text frame to the top of the slide. Read-only.

Class: word

Plural

words

Represents a word in a text range. Each item in the **words** list is a **text range** object that represents one word.

To return a **text range** object that represents one word, use **word** *index*, where *index* is the index number. The index number represents the position of the word in the **words** list. This example formats as bold the second, third, and fourth words in the first paragraph in shape two on slide one in the active presentation.

```
set theRange to text range of text frame of shape 2 of ¬
        slide 1 of active presentation
repeat with i from 2 to 4
    set italic of font of word i of theRange to true
end repeat
```

Properties

<*Inheritance*> **text range**

Inherits the properties and elements of the **text range** class.

Text Suite Commands

Command: add periods to

Adds a period at the end of each paragraph in the specified text.

Syntax

add periods to   text range   Required. An expression that returns a **text range** object.

Remarks

This command doesn't add another period at the end of a paragraph that already ends with a period.

Example

This example adds a period at the end of each paragraph in shape two on slide one in the active presentation.

```
set theRange to text range of text frame of shape 2 of slide 1 ¬
        of active presentation
add periods to theRange
```

Command: change case

Changes the case of the specified text.

Syntax

change case   text range   Required. An expression that returns a **text range** object.

  to   enumeration   Required. Specifies the way the case will be changed. Can be one of the following: **case lower**, **case sentence**, **case title**, or **case upper**.

Example

This example sets title case capitalization for the title on slide one in the active presentation.

```
set theRange to text range of text frame of shape 1 of slide 1 ¬
        of active presentation
change case theRange to case title
```

Command: `copy text range`

Copies the specified object to the Clipboard.

Syntax

copy text range   text range   Required. An expression that returns a **text range** object.

Remarks

To paste the contents of the Clipboard, use the **paste text range** command.

Example

This example copies the text in shape one on slide one in the active presentation to the Clipboard.

```
copy text range text range of text frame of shape 1 of slide 1 ¬
        of active presentation
```

Command: `cut text range`

Deletes the specified object and places it on the Clipboard.

Syntax

cut text range   text range   Required. An expression that returns a **text range** object.

Example

This example deletes the text in shape one on slide one in the active presentation and places a copy of it on the Clipboard.

```
cut text range text range of text frame of shape 1 of slide 1 ¬
        of active presentation
```

Command: `get text action setting`

Returns an **action setting** object that contains information about what action occurs when the user clicks or moves the mouse over the specified text range during a slide show.

Syntax

get text action setting for   text range   An expression that returns a **text range** object.

  result   enumeration   Required. The mouse action that triggers the action setting. Can be one of the following: **mouse activation mouse click** or **mouse activation mouse over**.

Example

The following example sets the actions for clicking and moving the mouse over shape one on slide two in the active presentation.

```
set myRange to text range of text frame of shape 1 of slide 2 ¬
        of active presentation
set mClick to get text action setting myRange ¬
        result mouse activation mouse click
set action of mClick to action type last slide
```

Command: `paste text range`

Pastes the text on the Clipboard into the specified text range.

Syntax

paste text range   text range   Required. An expression that returns a **text range** object.

Example

This example cuts the text in shape one on slide one in the active presentation, places it on the Clipboard, and then pastes it in shape two on the same slide.

```
set theSlide to slide 1 of active presentation
cut text range (text range of text frame of shape 1 of theSlide)
paste text range (text range of text frame of shape 2 of theSlide)
```

Command: `remove periods from`

Removes the period at the end of each paragraph in the specified text.

Syntax

remove periods from   text range   Required. An expression that returns a **text range** object.

Example

This example removes the period at the end of each paragraph in shape two on slide one in the active presentation.

```
remove periods from text range of text frame of shape 2 of slide 1 of ¬
        active presentation
```

Command: `rotate text bounds of`

This command is not currently supported.

Table Suite

Table Suite Classes

Class: cell

Plural

cells

Represents a table cell.

Use the **get cell from** command to return a single **cell** object. The following example creates a new slide and inserts a table, and then merges the first two cells in row one of the table.

```
set mySlide to make new slide at the beginning of active presentation ¬
        with properties {layout:slide layout blank}
set myShapeTable to make new shape table at the beginning of mySlide ¬
        with properties {number of rows:2, number of columns:3, ¬
        height:200, width:400}
set myTable to table object of myShapeTable
set c1 to get cell from myTable row 1 column 1
set c2 to get cell from myTable row 1 column 2
merge c1 merge with c2
```

Use the *shape* property to access the **shape** object and to manipulate the contents of each cell. This example deletes the text in the first cell (row 1, column 1), inserts new text, and then sets the width of the entire column to 110 points.

```
set myTable to table object of shape table 1 of slide 2 of active presentation
set c1 to get cell from myTable row 1 column 1
set content of text range of text frame of shape of c1 to "Rooster"
set width of column 1 of myTable to 110
```

Remarks

You cannot programmatically add cells to or delete cells from a PowerPoint table. Use the **make** command to add a column or row to a table. Use the **delete** command to delete a column or row from a table.

Properties

selected

Specifies whether the specified table cell is selected. Read-only.

shape

Returns a **shape** object that represents a shape in a table cell. Read-only.

Class: column

Plural

columns

Elements

cell

Represents a table column. The **columns** list includes all the columns in a table.

Use **column** *index* to return a single **column** object. *Index* represents the position of the column in the **column** list (usually counting from left to right; although the *table direction* property can reverse this). This example selects the first column of the table in shape one on the second slide.

```
select column 1 of table object of shape table 1 of slide 2 ¬
        of active presentation
```

Use the **cell** object to indirectly reference the **column** object. This example deletes the text in the first cell (row 1, column 1), inserts new text, and then sets the width of the entire column to 110 points.

```
set myTable to table object of shape table 1 of slide 2 of active presentation

set c1 to get cell from myTable row 1 column 1

set content of text range of text frame of shape of c1 to "Rooster"

set width of column 1 of myTable to 110
```

Use the **make** command to add a column to a table. This example creates a column in an existing table and sets the column width to 72 points (one inch).

```
set myTable to table object of shape table 1 of slide 2 of active presentation

make new column at the end of myTable with properties {width:72}
```

Properties

width

Returns or sets the width (in points) of the specified column. Read/write.

Class: row

Plural

rows

Elements

cell

Represents a row in a table. The **rows** list includes all the rows in the specified table.

Use **row** *index*, where *index* is a number that represents the position of the row in the table, to return a single **row** object. This example deletes the first row from the table in shape one on slide two of the active presentation.

```
set myTable to table object of shape table 1 of slide 2 of active presentation
delete row 1 of myTable
```

Properties

height

> Returns or sets the height (in points) of the specified row. Read/write.

Class: table

Plural

tables

Elements

column

row

Represents a table shape on a slide.

Use the *table object* property of the **shape table** object to return the table object from the shape containing the table.

To access the contents of each table cell, use the **get cell from** command. This example inserts the text "Cell 1" in the first cell of the table in shape five on slide three.

```
set myTable to table object of shape table 5 of slide 3 of active presentation
set c1 to get cell from myTable row 1 column 1
set content of text range of text frame of shape of c1 to "Cell 1"
```

Use the **make** command to add a table to a slide. This example adds a 3x3 table on slide two in the active presentation.

```
set theSlide to slide 2 of active presentation
set myTable to make new shape table at the beginning of theSlide ¬
        with properties {number of rows:3, number of columns:3}
```

Properties

table direction

Returns or sets the direction in which the table cells are ordered. Read/write.

Can be one of the following:

- **direction unset**
- **left to right**

Table Suite Commands

Command: `get border`

Returns a **line format** object that represents the borders and diagonal lines for the specified **cell** object.

Syntax

get border   cell   Required. An expression that returns a **cell** object.

> **edge**   enumeration   Required. The border that you want to return. Can be one of the following: **top border**, **left border**, **bottom border**, **right border**, **diagonal down border**, or **diagonal up border**.

Example

This example sets the thickness of the left border for the first cell in the second row of the specified table to three points.

```
set myTable to table object of shape table 1 of slide 1 of active presentation

set c1 to get cell from myTable row 2 column 1

set c1Border to get border c1 edge left border

set line weight of c1Border to 3
```

Command: `get cell from`

Returns a **cell** object that represents a cell in a table.

Syntax

get cell from   table   Required. An expression that returns a **table** object.

> **row**   integer   Required. The number of the row in the table to return. Can be an integer between 1 and the number of rows in the table.

> **column**   integer   Required. The number of the cell in the table to return. Can be an integer between 1 and the number of columns in the table.

Example

This example creates a 3x3 table on a new slide in a new presentation and inserts text into the first cell of the table.

```
make new presentation
set mySlide to make new slide at the beginning of active presentation ¬
        with properties {layout:slide layout blank}
set myShapeTable to make new shape table at the beginning of mySlide ¬
        with properties {number of rows:3, number of columns:3}
set myTable to table object of myShapeTable
set c1 to get cell from myTable row 1 column 1
set content of text range of text frame of shape of c1 to "Cell 1"
```

This example sets the thickness of the bottom border of the cell in row 2, column 1 to two points.

```
set myTable to table object of shape table 5 of slide 2 of active presentation
set c1 to get cell from myTable row 2 column 1
set theBorder to get border c1 edge bottom border
set line weight of theBorder to 2
```

Command: merge

Merges one table cell with another. The result is a single table cell.

Syntax

merge   cell   Required. An expression that returns a **cell** object.

  merge with   cell   Required. **Cell** object to be merged with.

Example

This example merges the first two cells of row one in the specified table.

```
set myTable to table object of shape table 5 of slide 2 of active presentation
set c1 to get cell from myTable row 1 column 1
set c2 to get cell from myTable row 1 column 2
merge c1 merge with c2
```

Command: `split`

Splits a single table cell into multiple cells.

Syntax

split   cell   Required. An expression that returns a **cell** object.

  number of rows   integer   Required. Number of rows that the cell is being split into.

  number of columns   integer   Required. Number of columns that the cell is being split into.

Example

This example splits the first cell in the referenced table into two cells, one directly above the other.

```
set myTable to table object of shape table 5 of slide 2 of active presentation
set c1 to get cell from myTable row 1 column 1
split c1 number of rows 2 number of columns 1
```