

ScriptUI Classes

Contents

JavaScript Reference [ScriptUI Classes](#) Guide

Scripting is one of the most powerful features in the CS3 suite. It can save lots of time: almost anything you can do with the user interface, you can do with scripts, and there are even a few things you can do in scripting that you cannot do from within the user interface. The contents on the right is categorized by function; in the index at the bottom, you can find all objects in alphabetical order. Each object description lists its elements, properties, and methods.

ScriptUI Classes

Contents

ScriptUI Classes

ScriptUI Classes

[Bounds](#)
[Button](#)
[Checkbox](#)
[Dimension](#)
[DrawState](#)
[DropDownList](#)
[EditText](#)
[FlashPlayer](#)
[Group](#)
[IconButton](#)
[LayoutManager](#)
[ListBox](#)
[ListItem](#)
[Panel](#)
[Point](#)
[Progressbar](#)
[RadioButton](#)
[ScriptUI](#)
[ScriptUIBrush](#)
[ScriptUIFont](#)
[ScriptUIGraphics](#)
[ScriptUIImage](#)
[ScriptUIPath](#)
[ScriptUIPen](#)
[Scrollbar](#)
[Slider](#)
[StaticText](#)
[TreeView](#)
[UIEvent](#)
[Window](#)

Class

Bounds

Defines the boundaries of a window within the screen's coordinate space, or of a UI element within the container's coordinate space.

A Bounds object is created when you set an element's `bounds` property. You can set the property using a JavaScript object with properties named `left`, `top`, `right`, `bottom` or `x`, `y`, `width`, `height`, or an array with 4 values in the order `[x, y, wd, ht]`.

Properties

Property	Type	Access	Description
<code>bottom</code>	Number	r/w	The vertical coordinate, a pixel offset from the origin of the element's coordinate space.
<code>height</code>	Number	r/w	The height in pixels.
<code>left</code>	Number	r/w	The horizontal coordinate, a pixel offset from the origin of the element's coordinate space.
<code>length</code>	Number	readonly	The array length. (value: 4)
<code>right</code>	Number	r/w	The width in pixels.
<code>top</code>	Number	r/w	The height in pixels.
<code>width</code>	Number	r/w	The width in pixels.
<code>x</code>	Number	r/w	The horizontal coordinate, a pixel offset from the origin of the element's coordinate space.
<code>y</code>	Number	r/w	The vertical coordinate, a pixel offset from the origin of the element's coordinate space.

Element of	<code>Button.bounds</code> <code>Button.windowBounds</code> <code>Checkbox.bounds</code> <code>Checkbox.windowBounds</code> <code>DropDownList.bounds</code> <code>DropDownList.windowBounds</code> <code>EditText.bounds</code> <code>EditText.windowBounds</code> <code>FlashPlayer.bounds</code> <code>FlashPlayer.windowBounds</code> <code>Group.bounds</code> <code>Group.windowBounds</code> <code>IconButton.bounds</code> <code>IconButton.windowBounds</code> <code>Image.bounds</code> <code>Image.windowBounds</code> <code>ListBox.bounds</code> <code>ListBox.windowBounds</code> <code>Panel.bounds</code> <code>Panel.windowBounds</code> <code>Progressbar.bounds</code> <code>Progressbar.windowBounds</code> <code>RadioButton.bounds</code> <code>RadioButton.windowBounds</code> <code>Scrollbar.bounds</code> <code>Scrollbar.windowBounds</code> <code>Slider.bounds</code> <code>Slider.windowBounds</code> <code>StaticText.bounds</code> <code>StaticText.windowBounds</code> <code>TreeView.bounds</code> <code>TreeView.windowBounds</code> <code>Window.bounds</code> <code>Window.frameBounds</code> <code>Window.windowBounds</code>
------------	---

Used in	<code>Group.add (type:String, bounds:Bounds, text:String, properties:Object)</code>
---------	---

[Panel.add](#) (*type:String*, *bounds:Bounds*, *text:String*,
properties:Object)
[Window.Window](#) (*type:String*, *title:String*, *bounds:Bounds*,
properties:Object)
[Window.add](#) (*type:String*, *bounds:Bounds*, *text:String*,
properties:Object)

[Contents](#) :: [Index](#)

Class

Button

A pushbutton element containing a mouse-sensitive text string.

Calls the [onClick](#) callback if the control is clicked or if its `notify()` method is called.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onClick](#), [onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	<p>True if this element is active.</p> <p>An active control is the one with keyboard focus—that is, the one that accepts keystrokes or in the case of <code>Button</code>, is selected when the user types Return or Enter in Windows, or the space bar in MacOS.</p>
<code>alignment</code>	<code>String</code>	r/w	<p>The alignment string for this element. If defined, this value overrides the <i>alignChildren</i> setting for the parent container.</p> <p>This can be a string, which indicates the alignment for the orientation speci</p>

in the parent container, or an array of two strings indicating both the horizontal and vertical alignment (in that order). Allowed values depend on the orientation value the parent container has. They are not case sensitive.

- For orientation=row: bottom, fill
- For orientation=column: left, right, fill
- For orientation=stack: bottom, left, right, fill

bounds	Bounds	r/w	<p>The boundaries of the element, in parent-relative coordinates.</p> <p>Setting an element size or location changes its bounds property, and vice versa.</p>
characters	Number	r/w	<p>A number of characters for which to reserve space when calculating preferred size of element.</p>

children	Array of Object	readonly	An array of child elements.
enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input, according to its type. When false, content elements do not accept input, and types of elements have a dimmed appearance.</p>
graphics	ScriptUIGraphics	readonly	The graphics object that can be used to customize the element's appearance, in response to the <code>onDraw</code> event.
helpTip	String	r/w	The help string that is displayed when the mouse hovers over the element.
indent	Number	r/w	<p>The number of pixels to indent the element during automatic layout.</p> <p>Applies for column orientation and left alignment, or row orientation and top alignment.</p>
justify	String	r/w	The text justification style. (default: left)

			One of left, center, right. Justification only works if this value is set on creation of the element.
location	Point	r/w	The upper left corner of this element relative its parent. The location is defined as [bounds.x, bounds.y]. Setting an element's location changes its bounds property, and vice versa.
maximumSize	Dimension	r/w	The maximum height and width which the element can be resized.
minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
parent	Object	readonly	The parent element.
preferredSize	Dimension	r/w	The preferred size used by layout managers to determine the best size for each element. If not explicitly set by a script, value

established by the UI framework in which ScriptUI is employed, and is based on such attributes of the element as its text font, font size, icon size, and other UI framework-specific attributes. A script can explicitly set preferredSize before the layout manager is invoked in order to establish an element size other than the default.

properties	Object	r/w	<p>An object that contains one or more creation properties of the container (properties used when the element is created).</p> <p>A Button object has no creation properties, but the third argument to the add() method that creates it can be an initial text value.</p>
shortcutKey	String	r/w	<p>The key sequence that invokes the onShortcutKey callback for this element (Windows only).</p>

size	Dimension	r/w	<p>The current dimensions of the element.</p> <p>Initially undefined and unless explicitly set by a script, it is defined by a <code>LayoutManager</code>. A script can explicitly set size before the layout manager is invoked to establish an element size other than the <i>preferredSize</i> or default size, but this is not recommended. Defined as <code>[bounds.width, bounds.height]</code>. Setting an element size changes its bounds property, vice-versa.</p>
text	String	r/w	The text to display as a localizable string.
type	String	readonly	The element type, "button".
visible	bool	r/w	<p>True if this element is shown, false if it is hidden.</p> <p>When a container is hidden, its children are also hidden, but they retain their own visibility values,</p>

			are shown or hid accordingly whe the parent is next shown.
window	Window	readonly	The window that this element belc to.
windowBounds	Bounds	readonly	The bounds of th element relative the top-level par window.

Methods **bool addEventListener** (*eventName:String*, *handler:Function* [, *capturePhase:bool=false*])

Registers an event handler for a particular type of event occurring this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be execute when the event occurs. A handler func takes one argument, the UIEvent object.
capturePhase	bool	When true, the handler is called only i the capturing phase of the event propagation. (default: false) Default is false, meaning that the handc is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.

[Event dispatchEvent](#) ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass it to this method to start the event propagation for the event.

void hide ()

Hides this element.

void notify (eventName:String)

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler to call. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . By default, this method simulates the <code>onChange</code> event for an edittext control, an <code>onClick</code> event for controls that support that event.

void onClick ()

An event-handler callback function, called when the element has been clicked

void onShortcutKey ()

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window.

In Windows only.

bool removeEventListener (eventName:String, handler:Function, capturePhase:bool=false)

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only in the capturing phase of the event propagation (default: false)

`void show ()`

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visibility states.

[Contents](#) :: [Index](#)

Class

Checkbox

A dual-state control showing a box that has a checkmark when the `value` is true, and is empty when the `value` is false.

Calls the `onClick` callback if the control is clicked or if its `onClick` method is called.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onClick](#), [onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	True if this checkbox is active. An active checkbox is the one with a checkmark. On keyboard, that is, the checkbox accepts keyboard input or in the context menu. On Button, is active when the checkbox is checked. On Return or Windows, space bar OS.
<code>alignment</code>	<code>String</code>	r/w	The alignment for this element. If defined, this overrides the <code>alignChild</code> setting for the container. This can be a string, which indicates the alignment.

orientation in the parent container, array of type indicating horizontal vertical alignment (in that order). Allowed values depend on orientation of the parent. They are not case sensitive.

- For orientation: top, bottom, fill
- For orientation: left, right, fill
- For orientation: top, bottom, left, right, fill

bounds	Bounds	r/w	The bounds of the element relative to the parent. Setting an size or location changes its property, and vice versa.
characters	Number	r/w	A number of characters to reserve when calculating preferred width.

			element.
children	Array of Object	readonly	An array of elements.
enabled	bool	r/w	True if this is enabled An enable can accept according When false elements can accept input types of elements have a default appearance
graphics	ScriptUIGraphics	readonly	The graphics that can be customized element's appearance response to onDraw event
helpTip	String	r/w	The help text displayed mouse hovering the element
indent	Number	r/w	The number of pixels to indent the element from the automatic alignment Applies for orientation alignment orientation alignment

justify	String	r/w	The default justification for child text (default: left). One of left, center, right. Justification only works if the value is set at the time of element creation.
location	Point	r/w	The upper-left corner of the element relative to its parent. The location is defined as [bounds.x, bounds.y]. If an element changes its location, the parent's location also changes, and vice versa.
maximumSize	Dimension	r/w	The maximum height and width to which the element can be resized.
minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
parent	Object	readonly	The parent element.
preferredSize	Dimension	r/w	The preferred height and width used by layout managers to determine the element's size.

size for each element.

If not explicitly defined by a script, the framework establishes a default UI framework which Screen employs based on screen attributes. For each element, attributes such as font, font size, and color are defined in the framework attributes. Developers can explicitly override these preferredScreenAttributes if the layout is invoked. The framework is invoked to establish the size of each element since the default size is defined in the framework.

properties	Object	r/w	An object contains a collection of creation parameters for the item (properties) when the item is created).
------------	--------	-----	---

A CheckBox has no creation parameters. The add() method creates the item to be displayed.

shortcutKey	String	r/w	The key stroke
-------------	--------	-----	----------------

			that invoke onShortcutKey for this element Windows
size	Dimension	r/w	The current dimension of the element. Initially undefined and unless set by a script set by a script defined by LayoutManager script can set size before layout manager invoked to an element other than <i>preferredSize</i> default size is not recommended Defined a [bounds.width bounds.height Setting an size change bounds property vice-versa
text	String	r/w	The text to localizable
type	String	readonly	The element "checkbox"
value	bool	r/w	The selected of the control When true control is

			selected o and displa check ma false, shov empty box
visible	bool	r/w	True if thi is shown, is hidden. When a co hidden, its are also hi they retain visibility v are shown according the parent shown.
window	Window	readonly	The windo this eleme to.
windowBounds	Bounds	readonly	The bound element re the top-lev window.

Methods **bool addEventListener** (*eventName:String*, *handler:Function*, *capturePhase:bool=false*)
Registers an event handler for a particular type of event on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>Jc Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function.

capturePhase bool

defined in the extension, or a locally defined handler function to be called when the event occurs. A handler takes one argument, the `UIEvent`

When true, the handler is called during the capturing phase of the event propagation. (default: **false**)
Default is false, meaning that the handler is called in the bubbling phase if the target object is an ancestor of the target object. The handler is called in the at-target phase if this object is the target.

Event `dispatchEvent ()`

Simulates the occurrence of an event in this target.

A script can create a `UIEvent` object for a specific event and pass it to this method to start the event propagation for the event.

`void hide ()`

Hides this element.

`void notify (eventName:String)`

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . <code>onChange</code> simulates the <code>onChange</code> event for an event control, an <code>onClick</code> event for controls that support that event.

`void onClick ()`

An event-handler callback function, called when the element has been clicked.

`void onShortcutKey ()`

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window.

In Windows only.

bool removeEventListener (*eventName:String*, *handler:*
capturePhase:bool=false)

Unregisters an event handler for a particular type of event in this element.

All arguments must be identical to those that were used to the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only during the capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visible states.

Class

Dimension

Defines the size of a window or UI element. Contains a 2-element array.

Specifies the height and width of an element in pixels. A Dimension object is created when you set an element's `size` property. You can set the property using a JavaScript object with properties named `width` and `height`, or an array with 2 values in the order [wd, ht].

Properties

Property	Type	Access	Description
<code>height</code>	Number	r/w	The height in pixels.
<code>length</code>	Number	readonly	The array length. (value: 2)
<code>width</code>	Number	r/w	The width in pixels.

Element of

[Button.maximumSize](#)
[Button.minimumSize](#)
[Button.preferredSize](#)
[Button.size](#)
[Checkbox.maximumSize](#)
[Checkbox.minimumSize](#)
[Checkbox.preferredSize](#)
[Checkbox.size](#)
[DropDownList.itemSize](#)
[DropDownList.maximumSize](#)
[DropDownList.minimumSize](#)
[DropDownList.preferredSize](#)
[DropDownList.size](#)
[EditText.maximumSize](#)
[EditText.minimumSize](#)
[EditText.preferredSize](#)
[EditText.size](#)
[FlashPlayer.maximumSize](#)
[FlashPlayer.minimumSize](#)
[FlashPlayer.preferredSize](#)

FlashPlayer.size
Group.maximumSize
Group.minimumSize
Group.preferredSize
Group.size
IconButton.maximumSize
IconButton.minimumSize
IconButton.preferredSize
IconButton.size
Image.maximumSize
Image.minimumSize
Image.preferredSize
Image.size
ListBox.itemSize
ListBox.maximumSize
ListBox.minimumSize
ListBox.preferredSize
ListBox.size
Panel.maximumSize
Panel.minimumSize
Panel.preferredSize
Panel.size
Progressbar.maximumSize
Progressbar.minimumSize
Progressbar.preferredSize
Progressbar.size
RadioButton.maximumSize
RadioButton.minimumSize
RadioButton.preferredSize
RadioButton.size
ScriptUIImage.size
Scrollbar.maximumSize
Scrollbar.minimumSize
Scrollbar.preferredSize
Scrollbar.size
Slider.maximumSize
Slider.minimumSize
Slider.preferredSize
Slider.size

[StaticText.maximumSize](#)
[StaticText.minimumSize](#)
[StaticText.preferredSize](#)
[StaticText.size](#)
[TreeView.itemSize](#)
[TreeView.maximumSize](#)
[TreeView.minimumSize](#)
[TreeView.preferredSize](#)
[TreeView.size](#)
[Window.frameSize](#)
[Window.maximumSize](#)
[Window.minimumSize](#)
[Window.preferredSize](#)
[Window.size](#)

Return **Dimension** [ScriptUIGraphics.measureString](#)
(*text:String*, *font:ScriptUIFont*,
boundingWidth:Number)

[Contents](#) :: [Index](#)

Class

DrawState

Describes an input state at the time of the triggering [ScriptUIGraphics.onDraw\(\)](#) event.

Contains properties that report whether the current control has the input focus, and the particular mouse button and keypress state. Passed in as argument to [ScriptUIGraphics.onDraw\(\)](#).

Properties

Property	Type	Access	Description
altKeyPressed	bool	readonly	True if the Alt key is being pressed (in Windows only).
capsLockKeyPressed	bool	readonly	True if the Caps Lock key is being pressed.
cmdKeyPressed	bool	readonly	True if the Command key is being pressed (in Mac OS only).
ctrlKeyPressed	bool	readonly	True if the Ctrl key is being pressed.
hasFocus	bool	readonly	True if the element has the input focus.

leftButtonPressed	bool	readonly	True if the left mouse button is being pressed.
middleButtonPressed	bool	readonly	True if the middle mouse button is being pressed.
mouseOver	bool	readonly	True if the cursor is hovering over this element.
numLockKeyPressed	bool	readonly	True if the Num Lock key is being pressed.
optKeyPressed	bool	readonly	True if the Option key is being pressed (in Mac OS only).
rightButtonPressed	bool	readonly	True if the right mouse button is being pressed.
shiftKeyPressed	bool	readonly	True if the Shift key is being pressed.

Used in [ScriptUIGraphics.onDraw](#) ([drawState:DrawState](#))

[Contents](#) :: [Index](#)

Class

DropDownList

Displays a single visible item. When you click drops down or pops up, and allows you to select items in the list.

Drop-down lists can have nonselectable separating groups of related items, as in a menu choice items on creation of the list object, or after the object's `add()` method. You can remove items from the list object's `remove()` and `removeAll()` methods. The `onChange` callback if the item selection is changed when the `select` method is called.

QuickLinks [add](#), [addEventListener](#), [dispatchEvent](#), [find](#), [highlight](#), [onShortcutKey](#), [remove](#), [removeAll](#), [removeEventListener](#)

Properties

Property	Type	Access
active	bool	r/w
alignment	String	r/w

bounds

Bounds

r/w

children	Array of Object	readonly
----------	-----------------	----------

enabled	bool	r/w
---------	------	-----

graphics	ScriptUIGraphics	readonly
----------	----------------------------------	----------

helpTip	String	r/w
---------	--------	-----

indent	Number	r/w
--------	--------	-----

itemSize

Dimension

r/w

items

Array of [ListItem](#) readonly

location	Point	r/w
----------	-------	-----

maximumSize	Dimension	r/w
-------------	-----------	-----

minimumSize	Dimension	r/w
-------------	-----------	-----

parent	Object	readonly
--------	--------	----------

preferredSize	Dimension	r/w
---------------	-----------	-----

properties

Object

r/w

selection

ListItem

r/w

shortcutKey

String

r/w

size

Dimension

r/w

type	String	readonly
------	--------	----------

visible	bool	r/w
---------	------	-----

window	Window	readonly
--------	--------	----------

windowBounds	Bounds	readonly
--------------	------------------------	----------

Methods [ListItem](#) **add** ([type:String](#), [text:String](#))

Adds an item or separator to the choices in this
Returns the item control object for type=item, (c
type=separator.

Parameter	Type	Description
type	String	The type of the child element Either <code>item</code> (a basic, select, label) or <code>separator</code>
text	String	The localizable text label

[bool](#) **addEventListener** ([eventName:String](#), [handler:Function](#), [capturePhase:bool=false](#))

Registers an event handler for a particular type
this element.

Parameter	Type	Description
eventName	String	The name of the event Event names are listed in <i>Tools Guide</i> .
handler	Function	The function that handles the event This can be the name of a function defined in the extension class or a function defined in the handler function when the event occurs takes one argument, the event object
capturePhase	bool	When true, the handler is called during the capturing phase of event propagation. (default is false, meaning bubbling)

is called in the bubble phase if the target object is an ancestor of the target. The at-target phase is the phase in which the target is the target.

Event **dispatchEvent** ()

Simulates the occurrence of an event in this target. A script can create a [UIEvent](#) object for a specific event and pass it to this method to start the event propagation for that event.

ListItem **find** (text:String)

Retrieves an item object from the list that has a text property equal to the specified text string.

Parameter	Type	Description
text	String	The text string to search for.

void **hide** ()

Hides this element.

void **notify** (eventName:String)

Sends a notification message, simulating the specified event. The event name must be one of the supported event names.

Parameter	Type	Description
eventName	String	The name of the control event. One of: <code>onClick</code> , <code>onChange</code> , or <code>onFocus</code> . The <code>onFocus</code> event simulates the <code>onFocus</code> event for a control, an <code>onClick</code> event for a button, and an <code>onChange</code> event for a text field. The <code>onFocus</code> event is not supported for all controls.

void **onChange** ()

An event-handler callback function, called when the value of the element has been changed.

void **onShortcutKey** ()

An event-handler callback function, called when the specified shortcutKey sequence is typed in the active window. In Windows only.

void **remove** (what:Any)

Removes a child item from the list.

Parameter	Type	Description
-----------	------	-------------

what	Any	The item or child to remove based index, text value, or
------	-----	---

void removeAll ()

Removes all child items from the list.

bool removeEventListener (eventName:String capturePhase:bool=false)

Unregisters an event handler for a particular type in this element.

All arguments must be identical to those that were used for the event handler.

Parameter	Type	Description
eventName	String	The name of the event
handler	Function	The function that handles the event
capturePhase	bool	Whether to call the handler during the capturing phase of the event (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are hidden but when it is shown again, the children retain their states.

Class

EditText

An editable text field that the user can select and change.

Calls the [onChange](#) callback if the text is changed and the user presses Enter or the control loses focus, or if its `notify()` method is called.

Calls the [onChanging](#) callback when any change is made to the text.
The `textselection` property contains currently selected text.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onChange](#), [onChanging](#), [onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	<p>True if this element is active.</p> <p>An active control is the one with keyboard focus; that is, the one that accepts keystrokes or in the case of a Button, is selected when the user presses Return or Enter in Windows, or space bar in Mac OS.</p>
<code>alignment</code>	<code>String</code>	r/w	<p>The alignment for this element. If defined, this overrides the <code>alignChildren</code> setting for the container.</p> <p>This can be a string, which indicates the</p>

alignment fo
orientation sp
in the parent
container, or
array of two
indicating bo
horizontal an
vertical align
(in that order
Allowed valu
depend on th
orientation v
the parent co
They are not
sensitive.

- For
orientation=r
bottom, fill
- For
orientation=c
left, right, fill
- For
orientation=s
bottom, left, righ

bounds	Bounds	r/w	The boundar the element, parent-relativ coordinates. Setting an el size or locati changes its b property, and versa.
characters	Number	r/w	A number of characters fo to reserve sp when calcula

			preferred size of the element.
children	Array of Object	readonly	An array of children elements.
enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input according to its type. When false, children elements do not accept input, and all types of elements have a dimmed appearance.</p>
graphics	ScriptUIGraphics	readonly	The graphics that can be used to customize the element's appearance, in response to the <code>onDraw</code> event.
helpTip	String	r/w	The help text displayed when the mouse hovers over the element.
indent	Number	r/w	<p>The number of pixels to indent the element during automatic layout.</p> <p>Applies for children orientation and alignment, or for orientation and alignment.</p>

justify	String	r/w	The text justify style. (default One of left, center, right. Justification only works if value is set on creation of the element.
location	Point	r/w	The upper left corner of this element relative to its parent. The location defined as [bounds.x, bounds.y]. Setting an element's location changes its bounds property, and vice versa.
maximumSize	Dimension	r/w	The maximum height and width which the element can be resized to.
minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
parent	Object	readonly	The parent element.
preferredSize	Dimension	r/w	The preferred size used by layout managers to determine the size for each element.

If not explicitly by a script, values are established by the UI framework which Script employed, or based on such attributes of the element as its font, font size, size, and other framework-specific attributes. A script can explicitly preferredSize the layout manager is invoked in to establish a element size than the default

properties

Object

r/w

An object that contains one creation property of the container (properties used when the element created).

Creation property of an EditText can include:

- multiline: When true (the default), control displays a single line of text. When true, the control displays multiple lines of text, which case the

wraps within
width of the

- `readonly`: Whether
(the default),
control accepts
input. When
the control does
accept input,
displays the
of the text pr
- `noecho`: Whether
(the default),
control displays
input text. When
true, the control
does not display its
text (used for
password input
fields).

-
`enterKeySignals`
When false (the
default), the control
signals an `onTextChanged`
event when the
editable text
changed and
control loses
keyboard focus,
the user tabs to
another control,
clicks outside
control, or types
Enter). When
the control is
signals an `onTextChanged`
event when the
editable text
changed and

			types Enter; it changes to the keyboard focus and does not signal the
shortcutKey	String	r/w	The key sequence that invokes the <code>onShortcutKeyClicked</code> for this element. Windows only
size	Dimension	r/w	The current dimensions of the element. Initially undefined and unless explicitly set by a script defined by a <code>LayoutManager</code> , a script can explicitly set size before layout management is invoked to establish an element's size other than the <i>preferredSize</i> default size, which is not recommended. Defined as <code>[bounds.width, bounds.height]</code> . Setting an element's size changes the bounds property vice-versa.
text	String	r/w	The current text displayed in the field, a localized string.

textselection	String	r/w	<p>The currently selected text, empty string is no text selection.</p> <p>Setting the value replaces the current text selection. If the text selection is modified, the value of the text property of the text widget is updated. If there is no current text selection, inserting a new value in the text string at the current insertion point. The textselection property is reset to an empty string after it modifies the value. Note that setting the textselection property before the widget's parent Window exists is an undefined operation.</p>
type	String	readonly	The element type "edittext".
visible	bool	r/w	<p>True if this widget is shown, false if hidden.</p> <p>When a container is hidden, its children are also hidden. They retain their visibility value.</p>

			are shown or accordingly v the parent is shown.
window	Window	readonly	The window this element to.
windowBounds	Bounds	readonly	The bounds of element relat the top-level window.

Methods **bool addEventListener** (*eventName:String*, *handler:Function* *capturePhase:bool=false*)

Registers an event handler for a particular type of event occur this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>Java Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a local defined handler function to be executed when the event occurs. A handler function takes one argument, the UIEvent object.
capturePhase	bool	When true, the handler is called on the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if the object is an ancestor of the target, the at-target phase if this object is the target.

Event dispatchEvent ()

Simulates the occurrence of an event in this target.
A script can create a `UIEvent` object for a specific event and to this method to start the event propagation for the event.

void hide ()

Hides this element.

void notify (eventName:String)

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . By default, <code>notify</code> simulates the <code>onChange</code> event for an edit control, an <code>onClick</code> event for controls that support that event.

void onChange ()

An event-handler callback function, called when the content element has been changed

The handler is called only when the change is complete—this is when focus moves to another control, or the user types Enter. The exact behavior depends on the creation parameter `enterKeySignalsOnChange`; see the `properties` property.

void onChanging ()

An event-handler callback function, called when the content element is in the process of changing

The handler is called for each keypress while this control has input focus.

void onShortcutKey ()

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window.

In Windows only.

bool removeEventListener (eventName:String, handler:Function, capturePhase:bool=false)

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only in the capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden, but when it is shown again, the children retain their own visibility states.

[Contents :: Index](#)

Class

FlashPlayer

A control that contains a Flash Player, which can load and play Flash movies stored in SWF files.

The ScriptUI FlashPlayer element runs the Flash application within an Adobe application. The Flash application runs ActionScript, a different implementation of JavaScript, the ExtendScript version of JavaScript that Adobe applications run. A control object of this type contains functions that your script to load SWF files, control movie playback, and communicate with the ActionScript environment.

QuickLinks [addEventListener](#), [callback](#), [dispatchEvent](#), [hide](#), [invokePlayerFunction](#), [loadMovie](#), [notify](#), [playMovie](#), [removeEventListener](#), [show](#), [stopMovie](#)

Properties

Property	Type	Access	Description
active	bool	r/w	True if this element is active. An active control is the one with keyboard focus; that is, the one that accepts keys or in the case of a Button, is selected when the user presses Return or Enter in Windows, or space bar in OS.
alignment	String	r/w	The alignment for this element. If defined, this overrides the <i>alignChildren</i>

setting for the container.

This can be a string, which indicates the alignment for orientation in the parent container, or an array of two indicating both horizontal and vertical alignment (in that order). Allowed values depend on the orientation value of the parent container. They are not case-sensitive.

- For orientation="vertical", allowed values are top, bottom, fill

- For orientation="horizontal", allowed values are left, right, fill

- For orientation="both", allowed values are bottom, left, right

bounds

Bounds

r/w

The boundary of the element, relative to its parent's coordinates.

Setting an element's size or location changes its bounds property, and

			versa.
enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input according to its type. When false, elements do not accept input, and all types of elements have a dimmed appearance.</p>
helpTip	String	r/w	The help text displayed when the mouse hovers over the element.
indent	Number	r/w	<p>The number of pixels to indent the element during automatic layout.</p> <p>Applies for column orientation and left alignment, or for row orientation and top alignment.</p>
location	Point	r/w	<p>The upper-left corner of this element relative to its parent.</p> <p>The location is defined as <code>[bounds.x, bounds.y]</code>. So when an element's location changes its bounds</p>

			property, and versa.
maximumSize	Dimension	r/w	The maximum height and width which the element can be resized.
minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
parent	Object	readonly	The parent element
preferredSize	Dimension	r/w	The preferred size used by layout managers to determine the size for each element. If not explicitly set by a script, the value is the established default for the UI framework which Script employed, as determined based on such attributes of the element as its font, font size, font color, and other framework-specific attributes. A script can explicitly set preferredSize to influence the layout manager when it is invoked to establish the preferred element size.

			than the defa
properties	Object	r/w	<p>An object th contains one creation prop of the contai (properties u when the ele created).</p> <p>A FlashPlay has no creati properties.</p>
size	Dimension	r/w	<p>The current dimensions o element.</p> <p>Initially und and unless e set by a scrip defined by a LayoutMana script can ex set size befo layout mana invoked to e an element s other than th <i>preferredSiz</i> default size, is not recom Defined as [bounds.wid bounds.heig] Setting an el size changes bounds prop vice-versa.</p>
type	String	readonly	The element

				"flashplayer"
visible	bool	r/w		True if this element is shown, false if hidden. When a container is hidden, its children are also hidden; they retain their visibility value and are shown or hidden accordingly when the parent is shown.
window	Window	readonly		The window to which this element belongs.
windowBounds	Bounds	readonly		The bounds of the element relative to the top-level window.

Methods `bool addEventListener (eventName:String, handler:Function, capturePhase:bool=false)`
Registers an event handler for a particular type of event occurring in this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function. The function will be executed when the event occurs. A handler function takes or

capturePhase bool

argument, the UIEvent object

When true, the handler is called only in the capturing phase of event propagation. (default: false) Default is false, meaning the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.

void callback ()

A function definition for a callback from the Flash ActionScript environment.

The Flash ActionScript code can call any callback function defined on the ExtendScript side of the FlashPlayer object by invoking it by name as a property of the control object. The function can take any arguments of a supported data type and can return any value of a supported data type. Data types: Number, String, Boolean, Null, undefined, Object, Array.

Event dispatchEvent ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass it to this method to start the event propagation for the event.

void hide ()

Hides this element.

Any invokePlayerFunction (name:String, argument:

Invokes an ActionScript function defined in the Flash application.

Returns the result of the invoked function, which must be of one of the allowed types. The ActionScript class and date objects are not supported as return values.

Parameter	Type	Description
name	String	The name of a Flash ActionScript function that has been registered.

the ExternalInterface object by 1 currently loaded SWF file.

argument Any An argument to pass through to function.
There can be any number of arg
An argument must be one of the types: Number, String, Boolean, undefined, Object, Array. No other types are supported.

void loadMovie (file:File)

Loads a movie into the Flash Player, and begins playing.

Parameter	Type	Description
file	File	The File object for the SWF file

void notify (eventName:String)

Sends a notification message, simulating the specified interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler to call. One of: <code>onClick</code> , <code>onChange</code> , <code>onChangeIn</code> default, simulates the <code>onChange</code> event for an edittext control, an <code>onClick</code> event for controls that support that event.

void playMovie ([rewind:bool=false])

Restarts a movie that has been stopped.

Do not use on a movie that is currently playing. The `stopMovie()-playMovie()` sequence does not work for files produced by Flex, or for some files produced by Authoring (depending on how they were implemented).

Parameter	Type	Description
rewind	bool	When true, restarts the movie from beginning; otherwise, starts playing from the point where it was stopped. (

false)

bool removeEventListener (*eventName:String*,
handler:Function [, *capturePhase:bool=false*])

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler during the capturing phase of the event propagation. (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden, but when it is shown again, the children retain their own visibility states.

void stopMovie ()

Halts playback of the current movie.

The stopMovie()-playMovie() sequence does not work for SWF files produced by Flex, or for some files produced using Flash Authoring (depending on how they were implemented). Using stopMovie() from the player's host environment has no effect on an SWF file playing in a ScriptUI Flash Player element. It is, however, possible to produce an SWF using Flash Authoring that can stop playback in response to user interaction.

Class

Group

A container for other controls within a window.

A group can specify layout options for its child elements. Hiding group hides all its children. Making it visible makes visible those children that are not individually hidden.

QuickLinks [add](#), [addEventListener](#), [dispatchEvent](#), [hide](#), [remove](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>alignChildren</code>	String	r/w	Tells the layout manager how unlike-sized children of this container should be aligned within a column or row. Order of creation determines which children are at the top of a column or the left of a row; earlier a child is created, the closer is to the top or left its column or row defined, <code>alignment</code> of a child element overrides the <code>alignChildren</code> setting for the parent container. See <code>alignment</code> property values.
<code>alignment</code>	String	r/w	The alignment style

for this element. If defined, this value overrides the *alignChildren* setting for the parent container.

This can be a single string, which indicates the alignment for the orientation specified in the parent container, or an array of two strings indicating both the horizontal and vertical alignment (in that order).

Allowed values depend on the orientation value of the parent container. They are not case sensitive.

- For `orientation=row;vertical=bottom`, fill

- For `orientation=column;vertical=left`, right, fill

- For `orientation=stack`, bottom, left, right, fill

bounds

Bounds

r/w

The boundaries of the element, in parent-relative coordinates.

			Setting an element size or location changes its bounce property, and vice versa.
children	Array of Object	readonly	An array of child elements.
enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input, according to its type. When false, control elements do not accept input, and other types of elements have a dimmed appearance.</p>
graphics	ScriptUIGraphics	readonly	The graphics object that can be used to customize the element's appearance, in response to the <code>onDraw</code> event.
helpTip	String	r/w	The help text that is displayed when the mouse hovers over the element.
indent	Number	r/w	<p>The number of pixels to indent the element during automatic layout.</p> <p>Applies for column orientation and left</p>

			alignment, or row orientation and to alignment.
layout	LayoutManager	r/w	<p>The layout manager for this container.</p> <p>The first time a container object is made visible, SwiftUI invokes this layout manager by calling its <code>layout</code> function. Default is an instance of the <code>LayoutManager</code> class that is automatically created when the container element is created.</p>
location	Point	r/w	<p>The upper left corner of this element relative to its parent.</p> <p>The location is defined as <code>[bounds.x, bounds.y]</code>. Setting an element's location changes its bounds property, and vice versa.</p>
margins	Number	r/w	The number of pixels between the edges of a container and the outermost child elements.

You can specify different margins each edge of the container. The default value is based on the type container, and is chosen to match the standard Adobe UI guidelines.

maximumSize	Dimension	r/w	The maximum height and width which the element can be resized.
-------------	-----------	-----	--

minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
-------------	-----------	-----	---

orientation	String	r/w	<p>The layout orientation of children in a container.</p> <p>Interpreted by the layout manager for the container. The default <code>LayoutManager</code> Object accepts the (case-insensitive) values <code>row</code>, <code>column</code>, <code>stack</code>. For window and panel, the default is <code>column</code> and for group the default is <code>row</code>. The allowed values for the container's</p>
-------------	--------	-----	---

alignChildren and its children's alignment properties depend on the orientation.

parent	Object	readonly	The parent element
preferredSize	Dimension	r/w	<p>The preferred size used by layout managers to determine the best size for each element.</p> <p>If not explicitly set by a script, value established by the UI framework in which ScriptUI is employed, and is based on such attributes of the element as its text font, font size, icon size, and other UI framework-specific attributes. A script can explicitly set preferredSize before the layout manager is invoked in order to establish an element size other than the default.</p>
properties	Object	r/w	An object that contains one or more creation properties of the control (properties used when the element

created).

A Group object has no creation properties.

size

Dimension

r/w

The current dimensions of this element.

Initially undefined and unless explicitly set by a script, it is defined by a `LayoutManager`. A script can explicitly set size before the layout manager is invoked to establish an element size other than the *preferredSize* or the default size, but this is not recommended. Defined as `[bounds.width, bounds.height]`. Setting an element size changes its bounds property, vice-versa.

spacing

Number

r/w

The number of pixels separating child element from its adjacent sibling element.

Because each container holds only a single row or

			column of children only a single space value is needed for container. The default value is based on the type of container, and is chosen to match standard Adobe UI guidelines.
type	String	readonly	The element type "group".
visible	bool	r/w	True if this element is shown, false if it is hidden. When a container is hidden, its children are also hidden, but they retain their own visibility values, and are shown or hidden accordingly when the parent is next shown.
window	Window	readonly	The window that this element belongs to.
windowBounds	Bounds	readonly	The bounds of this element relative to the top-level parent window.

Methods **Object add** (*type:String, bounds:Bounds, text:String, properties:Object*)
 Adds a child element to this container.
 Creates and returns a new control or container object and adds it

the children of this group.

Parameter	Type	Description
type	String	The type of the child element, as specified for the type property. Control types are listed in the <i>JavaScript Tools Guide</i> .
bounds	Bounds	A bounds specification that describes the size and position of the new control or container relative to its parent. If supplied, this value creates a new Bounds object which is assigned to the new object's bounds property.
text	String	The text or label, a localizable string. Initial text to be displayed in the control as the title, label, or contents, depending on the control type. If supplied, this value is assigned to the new object's text property.
properties	Object	An object that contains one or more creation properties of the new child (properties used only when the element is created). The creation properties depend on the element type. See properties property of each control type.

bool addEventListener (*eventName:String*, *handler:Function* [, *capturePhase:bool=false*])

Registers an event handler for a particular type of event occurring on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally

capturePhase bool

defined handler function to be executed when the event occurs. A handler function takes one argument, the UIEvent object

When true, the handler is called only in the capturing phase of the event propagation. (default: **false**)

Default is false, meaning that the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.

Event dispatchEvent ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass to this method to start the event propagation for the event.

void hide ()

Hides this element.

void remove (what:Any)

Removes the specified child control from this group's children array. No error results if the child does not exist.

Parameter	Type	Description
what	Any	The child control to remove, specified by 0-based index, text property value, or as a control object.

bool removeEventListener (eventName:String, handler:Function, capturePhase:bool=false)

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.

capturePhase	bool	Whether to call the handler only in the capturing phase of the event propagation (default: false)
--------------	------	---

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visibility states.

[Contents](#) :: [Index](#)

Class

IconButton

A mouse-sensitive pushbutton that displays an image in
Calls the [onClick](#) callback if the control is clicked or if
method is called.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onClick](#)
[onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Descri
active	bool	r/w	True if is activ An acti the one keyboa that is, accepts or in th Button, when tl Return Window space b OS.
alignment	String	r/w	The ali for this defined overrid <i>alignCi</i> setting contain This ca string, ' indicat alignm orienta

in the p
 contain
 array o
 indicati
 horizor
 vertical
 (in that
 Allowe
 depend
 orienta
 the par
 They a
 sensitiv
 - For
 orienta
 bottom, f
 - For
 orienta
 left, right
 - For
 orienta
 bottom, l

bounds	Bounds	r/w	The bo the ele parent- coordin Setting size or change propert versa.
children	Array of Object	readonly	An arra elemen
enabled	bool	r/w	True if is enab An ena

			can acc accordi When f elemen accept types o have a appear
graphics	ScriptUIGraphics	readonly	The gra that car custom elemen appear respons onDraw €
helpTip	String	r/w	The hel display mouse the ele
image	ScriptUIImage	r/w	The im that del image t
indent	Number	r/w	The nu pixels t elemen automa Applie orienta alignm orienta alignm
location	Point	r/w	The up corner elemen its pare

			The loc defined [bound bounds an elen change propert versa.
maximumSize	Dimension	r/w	The ma height ; which t can be :
minimumSize	Dimension	r/w	The mi and wic the eler resized
parent	Object	readonly	The pai
preferredSize	Dimension	r/w	The pre used by manage determi size for elemen If not e by a sci establis UI fran which ? employ based c attribut elemen font, fo size, an framew attribut

can exp
prefer
the lay
is invol
to estat
elemen
than th

properties	Object	r/w	An obj contain creation of the c (proper when tl created Creatio of an Ic object c - style: / the visu either " which l border or 3D a "toolbu has a fl appear approp inclusic toolbar
shortcutKey	String	r/w	The key that inv onShortc for this Window
size	Dimension	r/w	The cu dimens elemen

Initially and unl set by a defined Layout script c set size layout i invoked an elem other th *prefer* default is not r Defined [bound bounds Setting size ch bounds vice-ve

type	String	readonly	The ele "iconbu
visible	bool	r/w	True if is show is hidde When a hidden, are also they re visibili are sho accordi the par shown.
window	Window	readonly	The wi

			this element.
windowBounds	Bounds	readonly	The bounding box of the top-level window.

Methods `bool addEventListener (eventName:String, handler:Function, capturePhase:bool=false)`
Registers an event handler for a particular type of event on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a function defined handler function to be called when the event occurs. A handler function takes one argument, the UIEvent object.
capturePhase	bool	When true, the handler is called during the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase. If the target object is an ancestor of the target object, the handler is called in the at-target phase if this object is the target.

Event `dispatchEvent ()`

Simulates the occurrence of an event in this target. A script can create a [UIEvent](#) object for a specific event and pass it to this method to start the event propagation for the event.

`void hide ()`

Hides this element.

void notify (*eventName:String*)

Sends a notification message, simulating the specified interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> simulates the <code>onChange</code> event for a control, an <code>onClick</code> event for controls that support that event.

void onClick ()

An event-handler callback function, called when the element has been clicked.

void onShortcutKey ()

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window.

In Windows only.

bool removeEventListener (*eventName:String, handler:Function, capturePhase:bool=false*)

Unregisters an event handler for a particular type of event in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler during the capturing phase of the event (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are hidden but when it is shown again, the children retain their own states.

Contents :: Index

Class

Image

An element that displays an image.

An Image object has no creation properties, but the third argument of the `add()` method that creates it can be an icon, specified as a resource name, path, or File object for the image file. Images must be in PNG format.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>alignment</code>	String	r/w	<p>The alignment style for this element. If not defined, this value overrides the <i>alignChildren</i> setting for the parent container.</p> <p>This can be a single string, which indicates the alignment for the orientation specified in the parent container, or an array of two strings indicating both the horizontal and vertical alignment (in that order). Allowed values depend on the orientation value of the parent container. They are not case sensitive.</p>

			<ul style="list-style-type: none"> - For orientation=row:top, bottom, fill - For orientation=column:left, right, fill - For orientation=stack:bottom, left, right, fill
bounds	Bounds	r/w	<p>The boundaries of the element, in parent-relative coordinates.</p> <p>Setting an element's size or location changes its bounds property, and vice versa.</p>
children	Array of Object	readonly	An array of child elements.
enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input, according to its type. When false, control elements do not accept input, and other types of elements have a dimmed appearance.</p>
graphics	ScriptUIGraphics	readonly	The graphics object that can be used to customize the element's appearance, in

			response to the onDraw event.
helpTip	String	r/w	The help text that displayed when the mouse hovers over the element.
image	ScriptUIImage	r/w	The image object that defines the image to be drawn.
indent	Number	r/w	The number of pixels to indent the element during automatic layout. Applies for column orientation and left alignment, or row orientation and top alignment.
location	Point	r/w	The upper left corner of this element relative to its parent. The location is defined as [bounds.x, bounds.y]. Setting an element's location changes its bounds property, and vice versa.
maximumSize	Dimension	r/w	The maximum height and width which the element can be resized.

minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
parent	Object	readonly	The parent element
preferredSize	Dimension	r/w	<p>The preferred size used by layout managers to determine the best size for each element.</p> <p>If not explicitly set by a script, value established by the UI framework in which ScriptUI is employed, and is based on such attributes of the element as its text font, font size, icon size, and other UI framework-specific attributes. A script can explicitly set preferredSize before the layout manager is invoked in order to establish an element size other than the default.</p>
properties	Object	r/w	An object that contains one or more creation properties of the container (properties used when the element

created).

An Image object has no creation properties, but the third argument to add() method that creates it can be a icon, specified as resource name, package or File object for image file. Image must be in PNG format.

size

Dimension

r/w

The current dimensions of this element.

Initially undefined and unless explicitly set by a script, it is defined by a LayoutManager. A script can explicitly set size before the layout manager is invoked to establish an element size other than the *preferredSize* or the default size, but this is not recommended. Defined as [bounds.width, bounds.height]. Setting an element size changes its bounds property, vice-versa.

type	String	readonly	The element type "image".
visible	bool	r/w	True if this element is shown, false if is hidden. When a container hidden, its children are also hidden, but they retain their own visibility values, and are shown or hidden accordingly when the parent is next shown.
window	Window	readonly	The window that this element belongs to.
windowBounds	Bounds	readonly	The bounds of this element relative to the top-level parent window.

Methods `bool addEventListener (eventName:String, handler:Function [, capturePhase:bool=false])`
Registers an event handler for a particular type of event occurring on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be executed when the event occurs. A handler function

capturePhase	bool	takes one argument, the UIEvent object When true, the handler is called only in the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.
--------------	------	---

Event [dispatchEvent](#) ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass to this method to start the event propagation for the event.

void [hide](#) ()

Hides this element.

bool [removeEventListener](#) (*eventName:String*, *handler:Function*, *capturePhase:bool=false*)]

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only in the capturing phase of the event propagation (default: false)

void [show](#) ()

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visibility states.

Class

LayoutManager

Controls the automatic layout behavior for a window or container.

The subclass `AutoLayoutManager` implements the default automatic layout behavior.

QuickLinks [layout](#), [resize](#)

Methods `void layout ()`

Invokes the automatic layout behavior for the managed container.

Adjusts sizes and positions of the child elements of this window or container according to the placement and alignment property values in the parent and children. Invoked automatically the first time the window is displayed. Thereafter, the script must invoke it explicitly to change the layout in case of changes in the size or position of the parent or children.

`void resize ()`

Performs a layout after a Window is resized, based on the new size.

Resizes the child elements of the managed container with a given alignment type, after the window has been resized by the user.

Element of [Group.layout](#)
[Panel.layout](#)
[Window.layout](#)

[Contents](#) :: [Index](#)

Class

ListBox

Displays a list of choices, represented by [ListItem](#) objects.

When you create the object, you specify whether it allows the select only one or multiple items. If a list contains more items be displayed in the available area, a scrollbar may appear that user to scroll through all the list items. You can specify the cho on creation of the list object, or afterward using the list object' method. You can remove items programmatically with the list `remove()` and `removeAll()` methods.

QuickLinks [add](#), [addEventListener](#), [dispatchEvent](#), [find](#), [hide](#), [notify](#), [onCh](#), [onShortcutKey](#), [remove](#), [removeAll](#), [removeEventListener](#), [sh](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	True if this ele active. An active cont one with keybo focus—that is, that accepts ke or in the case c Button, is selec when the user Return or Ente Windows, or tl bar in Mac OS
<code>alignment</code>	<code>String</code>	r/w	The alignment this element. If this value over <i>alignChildren</i> for the parent c This can be a s string, which i the alignment t orientation spe

the parent contains an array of two values indicating both horizontal and vertical alignment (in that order). Allowed values depend on the orientation value the parent contains. Values are not case sensitive.

- For orientation=rotated: top, bottom, fill
- For orientation=column: left, right, fill
- For orientation=stacked: top, bottom, left, right, fill

bounds	Bounds	r/w	The boundaries of the element, in parent relative coordinates. Setting an element's size or location changes its bounds proportionally, vice-versa.
--------	------------------------	-----	---

children	Array of Object	readonly	An array of child ListItem objects.
----------	-----------------	----------	---

enabled	bool	r/w	True if this element is enabled. An enabled element will accept input, according to its type. When disabled, a control element will not accept input, and some types of elements will not be visible.
---------	------	-----	---

			a dimmed appearance
graphics	ScriptUIGraphics	readonly	The graphics object can be used to customize the appearance, in response to the <code>onDraw</code> event.
helpTip	String	r/w	The help text that is displayed when the mouse hovers over the element.
indent	Number	r/w	The number of pixels to indent the element during automatic layout. Applies for column orientation and alignment, or row orientation and alignment.
itemSize	Dimension	r/w	The width and height in pixels of each item in the list. Used by auto-layout to determine the <code>preferredSize</code> of the items, if not otherwise specified. If not set explicitly, the size of each item is determined to match the layout height and width of all items in the list.
items	Array of ListItem	readonly	The array of class objects representing the items displayed in the list. Access this array to modify the items in the list.

			0-based index. obtain the number of items in the list using <code>items.length</code> . The items are created when they are specified on the creation of the list object, or added using the list <code>add()</code> method. Each item has a <code>selected</code> property that is <code>true</code> when it is in the selected state.
<code>location</code>	Point	r/w	The upper left corner of this element relative to its parent. The location is specified as <code>[bounds.x, bounds.y]</code> . Setting an element's <code>location</code> changes its <code>bounds</code> property, and vice versa.
<code>maximumSize</code>	Dimension	r/w	The maximum height and width to which an element can be stretched.
<code>minimumSize</code>	Dimension	r/w	The minimum height and width to which an element can be stretched.
<code>parent</code>	Object	readonly	The parent element.
<code>preferredSize</code>	Dimension	r/w	The preferred size determined by layout managers. Layout managers determine the preferred size for each element.

If not explicitly script, value is established by framework in v ScriptUI is em and is based on attributes of th as its text, font size, icon size, other UI frame specific attribu script can expl preferredSize l layout manage invoked in ord establish an ele size other than default.

properties	Object	r/w	
			<p>An object that one or more cr properties of th (properties use when the elem created).</p> <p>Creation prope ListBox object include:</p> <ul style="list-style-type: none"> - multiselect: Wh (the default), o item can be sel When true, mu items can be se - items: An arra strings for the each list item. object is create each item. An the text string

creates a separate
Supply this property
the *items* argument
the add() method
both. This form is
useful for elements
defined using the
Specifications.

selection

ListItem

r/w

The currently selected
item for a single
selection list, or
array of items for
current selection
multi-selection.

Setting this value
causes the selected
to be highlighted
to be scrolled into
view if necessary. If no
items are selected, this
property is null. Set to
true to reselect all
items. You can set the
value to the index of
an item in an array of
indices rather than
object references. If
selected is an index
value that is outside
of range, the object
reference is ignored. When
selected is an index
value, the selected
property still returns
the object reference.
- If you set the
selected property to
an array for a
selection list, the
first item in the
array is selected.

			- If you set the a single item for multi-selection item is added to current selection
shortcutKey	String	r/w	The key sequence invokes the <code>onSelect</code> callback for this element (in Win only).
size	Dimension	r/w	The current dimension of this element Initially undefined unless explicitly a script, it is defined as a <code>LayoutManager</code> script can explicitly size before the manager is invoked establish an element size other than <i>preferredSize</i> as default size, but not recommended Defined as <code>[bounds.width, bounds.height]</code> an element's size changes its <code>bounds</code> property, and vice versa.
type	String	readonly	The element type "listbox".
visible	bool	r/w	True if this element is shown, false if hidden.

When a container is hidden, its children are also hidden, but children retain their own visibility value. Children are shown or hidden accordingly when the parent is next shown.

window	Window	readonly	The window that contains this element belongs to.
windowBounds	Bounds	readonly	The bounds of this element relative to the top-level parent window.

Methods [ListItem](#) **add** (*type:String*, *text:String*)
 Adds an item to the choices in this list.
 Returns the item control object.

Parameter	Type	Description
type	String	The type of the child element, the string value of the <code>type</code> property.
text	String	The localizable text label for the item.

bool [addEventListener](#) (*eventName:String*, *handler:Function*, *capturePhase:bool=false*)
 Registers an event handler for a particular type of event occurring on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function on the extension, or a locally defined handler function to be executed when the event occurs. A handler function takes one or more arguments.

capturePhase	bool	argument, the UIEvent object. When true, the handler is called only capturing phase of the event propagation (default: false) Default is false, meaning that the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.
--------------	------	--

Event dispatchEvent ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass it to this method to start the event propagation for the event.

ListItem find (text:String)

Retrieves an item object from the list that has a given text label.

Parameter	Type	Description
text	String	The text string to match.

void hide ()

Hides this element.

void notify (eventName:String)

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler to call. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . By default, this method simulates the <code>onChange</code> event for an edited control, an <code>onClick</code> event for controls that have a click event, and that event.

void onChange ()

An event-handler callback function, called when the content of the element has been changed.

void onShortcutKey ()

An event-handler callback function, called when the element's shortcut key sequence is typed in the active window.

In Windows only.

void remove (*what:Any*)

Removes a child item from the list.

Parameter	Type	Description
what	Any	The item or child to remove, specified by index, text value, or as a ListItem object.

void removeAll ()

Removes all child items from the list.

bool removeEventListener (*eventName:String, handler:Func capturePhase:bool=false*)

Unregisters an event handler for a particular type of event occurring on this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only in the capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden. When it is shown again, the children retain their own visibility.

Class

ListItem

A choice item in a list box, drop-down list, or tree view.

You can specify initial items in the creation parameters when creating the parent list. Create new items using the `add()` method in the parent list with control type="item", or, for `DropDownList` controls, type="separator".

Properties	Property	Type	Access	Description
	checked	bool	r/w	The checked state of an item in a list. When true, the item is marked with the platform-appropriate checkmark. When false, no checkmark is drawn, but space is reserved for it in the left margin, so that the item lines up with other checkable items. When undefined, no space is reserved for a checkmark.
expanded	bool	r/w	The expansion state of an item of type <i>node</i> that is a child of a TreeView list control. When true, the	

			item is in the expanded state and its children are shown, when false, it is collapsed and children are hidden.
image	ScriptUIImage	r/w	An image object for an icon to display in the item. When specified, the image appropriate to the selections state is drawn to the left of the text label.
index	Number	readonly	The 0-based index of this item in the <i>items</i> collection of its parent list control.
parent	Object	readonly	The parent element, a list control.
properties	Object	r/w	An object that contains one or more creation properties of the item (properties used only when the element is created). A ListItem object has no creation properties.

selected	bool	r/w	<p>The selection state of this item.</p> <p>When true, the item is part of the selection for its parent list. When false, the item is not selected. Set to true to select this item in a single-selection list, or to add it to the selection array for a multi-selection list.</p>
text	String	r/w	<p>The label text to display for the item, a localizable string.</p>
type	String	readonly	<p>The element type.</p> <p>Normally "item", but an item whose parent is a DropDownList control can have type "separator". A separator item is not mouse-sensitive and is drawn as a horizontal line across the drop-down or pop-up menu.</p>

Element of [DropDownList.items](#)

[DropDownList.selection](#)
[ListBox.items](#)
[ListBox.selection](#)
[TreeView.items](#)
[TreeView.selection](#)

Return **Listitem** [DropDownList.add](#) ([type:String](#), [text:String](#))
Listitem [DropDownList.find](#) ([text:String](#))
Listitem [ListBox.add](#) ([type:String](#), [text:String](#))
Listitem [ListBox.find](#) ([text:String](#))
Listitem [TreeView.add](#) ([type:String](#), [text:String](#))
Listitem [TreeView.find](#) ([text:String](#))

[Contents :: Index](#)

Class

Panel

A container for other types of controls, with an optional frame.

A panel can specify layout options for its child elements. Hiding panel hides all its children. Making it visible makes visible those children that are not individually hidden.

QuickLinks [add](#), [addEventListener](#), [dispatchEvent](#), [hide](#), [remove](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>alignChildren</code>	String	r/w	Specifies how to align the child elements.
<code>alignment</code>	String	r/w	<p>The alignment style for this element. If defined, this value overrides the <i>alignChildren</i> setting for the parent container.</p> <p>This can be a single string, which indicates the alignment for the orientation specified in the parent container, or an array of two strings indicating both the horizontal and vertical alignment (in that order). Allowed values depend on the orientation value</p>

the parent contain
They are not case sensitive.

- For orientation=row:top, bottom, fill

- For orientation=column:left, right, fill

- For orientation=stack:bottom, left, right, fill

bounds	Bounds	r/w	<p>The boundaries of the element, in parent-relative coordinates.</p> <p>Setting an element's size or location changes its bounds property, and vice versa.</p>
characters	Number	r/w	Reserve space for the specified number of characters; affects calculation of preferredSize.
children	Array of Object	readonly	An array of child elements.
enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input, according to its type. When false, control elements do not accept input, and</p>

			types of elements have a dimmed appearance.
graphics	ScriptUIGraphics	readonly	The graphics object that can be used to customize the element's appearance, in response to the <code>onDraw</code> event.
helpTip	String	r/w	The help text that displayed when the mouse hovers over the element.
indent	Number	r/w	The number of pixels to indent the element during automatic layout. Applies for column orientation and left alignment, or row orientation and top alignment.
justify	String	r/w	The default text justification style for child text elements (default: left) One of <code>left</code> , <code>center</code> , <code>right</code> , or <code>justify</code> . Justification only works if this value is set on creation of the element.
layout	LayoutManager	r/w	The layout manager for this container.

			<p>The first time a container object is made visible, SwiftUI invokes this layout manager by calling its <code>layout</code> function. Default is an instance of the <code>LayoutManager</code> class that is automatically created when the container element is created.</p>
location	Point	r/w	<p>The upper left corner of this element's frame relative to its parent.</p> <p>The location is defined as <code>[bounds.x, bounds.y]</code>. Setting an element's location changes its bounds property, and vice versa.</p>
margins	Number	r/w	<p>The number of pixels between the edges of a container and the outermost child elements.</p> <p>You can specify different margins for each edge of the container. The default value is based on the type</p>

			container, and is chosen to match the standard Adobe UI guidelines.
maximumSize	Dimension	r/w	The maximum height and width which the element can be resized.
minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
orientation	String	r/w	<p>The layout orientation of children in a container.</p> <p>Interpreted by the layout manager for the container. The default <code>LayoutManager</code> <code>Object</code> accepts the (case-insensitive) values <code>row</code>, <code>column</code>, <code>stack</code>. For <code>window</code> and <code>panel</code>, the default is <code>column</code> and for <code>group</code> the default is <code>row</code>. The allowed values for the container's <code>alignChildren</code> and its children's <code>alignment</code> properties depend on the orientation.</p>
parent	Object	readonly	The parent element

preferredSize	Dimension	r/w	<p>The preferred size used by layout managers to determine the best size for each element.</p> <p>If not explicitly set by a script, value established by the UI framework in which ScriptUI is employed, and is based on such attributes of the element as its text font, font size, icon size, and other UI framework-specific attributes. A script can explicitly set preferredSize before the layout manager is invoked in order to establish an element size other than the default.</p>
properties	Object	r/w	<p>An object that contains one or more creation properties of the control (properties used when the element is created).</p> <p>Creation properties of a Panel object include:</p> <ul style="list-style-type: none"> - borderStyle: A string

that specifies the appearance of the border drawn around the panel. One of black, etched, gray, raised, sunken. Default is etched.

- su1PanelCoordinates

Photoshop only.

When true, this panel automatically adjusts the position of its children for compatibility with Photoshop CS.

Default is false, meaning that the panel does not adjust the positions of its children, even if the parent window has automatic adjustment enabled.

size

Dimension

r/w

The current dimensions of this element.

Initially undefined and unless explicitly set by a script, it is defined by a

LayoutManager.

A script can explicitly set size before the layout manager is invoked to establish an element size

other than the *preferredSize* or *t*

			<p>default size, but t is not recommend Defined as [bounds.width, bounds.height]. Setting an elemer size changes its bounds property, vice-versa.</p>
spacing	Number	r/w	<p>The number of pixels separating child element from its adjacent siblin element.</p> <p>Because each container holds o a single row or column of childre only a single spac value is needed fo container. The default value is based on the type container, and is chosen to match standard Adobe U guidelines.</p>
text	String	r/w	The title or label text, a localizable string.
type	String	readonly	The element type "panel".
visible	bool	r/w	<p>True if this eleme is shown, false if is hidden.</p> <p>When a container</p>

hidden, its children are also hidden, but they retain their original visibility values, and are shown or hidden accordingly when the parent is next shown.

window	Window	readonly	The window that this element belongs to.
windowBounds	Bounds	readonly	The bounds of this element relative to the top-level parent window.

Methods [Object add](#) (*type:String, bounds:Bounds, text:String, properties:Object*)

Adds a child element to this container.

Creates and returns a new control or container object and adds it to the children of this group.

Parameter	Type	Description
type	String	The type of the child element, as specified for the type property. Control types are listed in the <i>JavaScript Tools Guide</i> .
bounds	Bounds	A bounds specification that describes the size and position of the new control or container relative to its parent. If supplied, this value creates a new Bounds object which is assigned to the new object's <code>bounds</code> property.
text	String	The text or label, a localizable string. Initial text to be displayed in the control as the title, label, or contents, depending on the control type. If supplied, this value is

properties	Object	assigned to the new object's <code>text</code> property. An object that contains one or more creation properties of the new child (properties used only when the element is created). The creation properties depend on the element type. See <code>properties</code> property of each control type.
------------	--------	---

bool addEventListener (`eventName:String`, `handler:Function` [, `capturePhase:bool=false`])

Registers an event handler for a particular type of event occurring on this element.

Parameter	Type	Description
<code>eventName</code>	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
<code>handler</code>	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be executed when the event occurs. A handler function takes one argument, the <code>UIEvent</code> object.
<code>capturePhase</code>	bool	When true, the handler is called only in the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.

Event dispatchEvent ()

Simulates the occurrence of an event in this target.

A script can create a `UIEvent` object for a specific event and pass it to this method to start the event propagation for the event.

void hide ()

Hides this element.

void remove (*what:Any*)

Removes the specified child control from this group's `children` array.
No error results if the child does not exist.

Parameter	Type	Description
<code>what</code>	<code>Any</code>	The child control to remove, specified by 0-based index, text property value, or as a control object.

bool removeEventListener (*eventName:String, handler:Function, capturePhase:bool=false*)

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
<code>eventName</code>	<code>String</code>	The name of the event.
<code>handler</code>	<code>Function</code>	The function that handles the event.
<code>capturePhase</code>	<code>bool</code>	Whether to call the handler only in the capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visibility states.

Class

Point

Defines the location of a window or UI element. Contains a 2-element array.

Specifies the origin point of an element as horizontal and vertical pixel offsets from the origin of the element's coordinate space. A Point object is created when you set an element's `location` property. You can set the property using a JavaScript object with properties named `x` and `y`, or an array with 2 values in the order `[x, y]`.

Properties

Property	Type	Access	Description
<code>left</code>	Number	r/w	The left coordinate.
<code>length</code>	Number	readonly	The array length. (value: 2)
<code>top</code>	Number	r/w	The top coordinate.
<code>x</code>	Number	r/w	The horizontal coordinate, a pixel offset from the origin of the element's coordinate space.
<code>y</code>	Number	r/w	The vertical coordinate, a pixel offset from the origin of the element's coordinate space.

Element of

- [Button.location](#)
- [Checkbox.location](#)
- [DropDownList.location](#)
- [EditText.location](#)
- [FlashPlayer.location](#)
- [Group.location](#)
- [IconButton.location](#)
- [Image.location](#)
- [ListBox.location](#)
- [Panel.location](#)
- [Progressbar.location](#)
- [RadioButton.location](#)
- [ScriptUIGraphics.currentPoint](#)
- [Scrollbar.location](#)

[Slider.location](#)
[StaticText.location](#)
[TreeView.location](#)
[Window.frameLocation](#)
[Window.location](#)

Return **Point** [ScriptUIGraphics.ellipsePath](#) ([left:Number](#), [top:Number](#),
[width:Number](#), [height:Number](#))
Point [ScriptUIGraphics.lineTo](#) ([x:Number](#), [y:Number](#))
Point [ScriptUIGraphics.moveTo](#) ([x:Number](#), [y:Number](#))
Point [ScriptUIGraphics.rectPath](#) ([left:Number](#), [top:Number](#),
[width:Number](#), [height:Number](#))

[Contents](#) :: [Index](#)

Class

Progressbar

A horizontal bar with an indicator that shows the progress of an operation.

All progressbar controls have a `horizontal` orientation property. The `value` property contains the current position of the progress indicator. The default is 0. There is a `minvalue` property, but it is always set to a different value and is silently ignored.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [removeEventListener](#)

Properties

Property	Type	Access	Description
<code>alignment</code>	String	r/w	The alignment of the progress indicator. The default is <code>center</code> . This property is only used for vertical progress indicators. The <code>align</code> property of the <code>ProgressIndicator</code> class is deprecated. They

			sensit - For orient bottom - For orient left, rig - For orient bottom
bounds	Bounds	r/w	The b the el paren coord Settir size c chang prope versa
children	Array of Object	readonly	An ar eleme
enabled	bool	r/w	True is ena An er can a accor When eleme accep types have appea
graphics	ScriptUIGraphics	readonly	The g that c custo eleme

			appears response onDraw
helpTip	String	r/w	The help display mouse the el
indent	Number	r/w	The number pixels element auto Applies orientation alignment orientation alignment
location	Point	r/w	The location corner element its position The location definition [boundaries] boundaries an element change properties versus
maximumSize	Dimension	r/w	The maximum height which can be
maxvalue	Number	r/w	The maximum in the is 100

minimumSize	Dimension	r/w	The n and w the el resize
minvalue	Number	r/w	The n in the 0. If s value
parent	Object	readonly	The p
preferredSize	Dimension	r/w	The p used l mana deteri size f eleme If not by a s establ UI fra which empla based attrib eleme font, : size, : frame attrib can e prefer the la is inv to est eleme than t
properties	Object	r/w	An ol

conta
creati
of the
(prop
when
create

A Proc
objec
creati
The t
of the
that c
initial
0), an
argun
maxi
the ra
100).

size

Dimension

r/w

The c
dimer
elem

Initia
and u
set by
defin
Layo
script
set si
layou
invok
an ele
other
prefe
defau
is not
Defin
[bour

			bound Setting size c bound vice-
type	String	readonly	The e "prog
value	Number	r/w	The c of the If set outsid speci: and m auton to the bound
visible	bool	r/w	True is shc is hid When hidde are al they r visibi are sh accor the pa show
window	Window	readonly	The v this e to.
windowBounds	Bounds	readonly	The b elemc the to windc

Methods **bool addEventListener** (*eventName:String*, *handler:capturePhase:bool=false*])

Registers an event handler for a particular type of event on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a defined handler function to be called when the event occurs. A handler takes one argument, the UIEvent object.
capturePhase	bool	When true, the handler is called during the capturing phase of the event propagation. (default: false) Default is false, meaning the handler is called in the bubbling phase if the target object is an ancestor of the event target.

Event dispatchEvent ()

Simulates the occurrence of an event in this target. A script can create a **UIEvent** object for a specific event and pass it to this method to start the event propagation for the event.

void hide ()

Hides this element.

bool removeEventListener (*eventName:String*, *handler:capturePhase:bool=false*])

Unregisters an event handler for a particular type of event on this element.

All arguments must be identical to those that were used when the event handler was registered.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles t
capturePhase	bool	Whether to call the handler capturing phase of the event (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are hidden but when it is shown again, the children retain their current states.

[Contents](#) :: [Index](#)

Class

RadioButton

A dual-state control, grouped with other radiobutton one can be in the selected state.

Shows the selected state when value=true, empty w
Calls the [onClick](#) callback if the control is clicked c
method is called.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onClick](#), [onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Des
active	bool	r/w	True is ac An the keyl that acce or in Butt whe Retu Win spac OS.
alignment	String	r/w	The for t defi over <i>alig</i> setti cont This stir indi

align
orient
in the
cont
array
indi
hori
vert
(in t
Alloc
depe
orient
the j
The
sens
- Fo
orient
botto
- Fo
orient
left, i
- Fo
orient
botto

bounds	Bounds	r/w	The the c pare cool Sett size char prop vers
--------	---------------	-----	--

characters	Number	r/w	A n char to re whe
------------	--------	-----	-----------------------------

				pref elen
children	Array of Object	readonly	An	elen
enabled	bool	r/w	True is en	An can acce Wh elen acce type havi app
graphics	ScriptUIGraphics	readonly	The that cust elen app resp onDr	
helpTip	String	r/w	The disp mou the	
indent	Number	r/w	The pixe elen autc	App orie alig orie alig

justify	String	r/w	The justify child (default) One right. only value created element
location	Point	r/w	The correct element its position The default [bound] bounding an element character property version
maximumSize	Dimension	r/w	The height which can
minimumSize	Dimension	r/w	The and the resizing
parent	Object	readonly	The
preferredSize	Dimension	r/w	The user manual details

size
elen
If no
by a
esta
UI f
whi
emp
base
attri
elen
font
size
fran
attri
can
pref
the
is in
to e
elen
thar

properties	Object	r/w	An o cont crea of tl (pro whe crea A R obje crea The of tl that the
shortcutKey	String	r/w	The

			that onSh for t Win
size	Dimension	r/w	The dim elen Initi and set l defi Lay scrij set s layc invc an e othe <i>pref</i> defa is no Def [bou bou Sett size bou vice
text	String	r/w	The this loca
type	String	readonly	The "rad
value	bool	r/w	The of tl sele

visible	bool	r/w	True is shown, false is hidden. When hidden, the element is not accessible to the user.
window	Window	readonly	The window of this element.
windowBounds	Bounds	readonly	The bounds of the window of this element.

Methods **bool addEventListener** (*eventName:String*, *handler:Function*, *capturePhase:bool=false*)

Registers an event handler for a particular type of event on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in <i>Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a defined handler function that is called when the event occurs. A function takes one argument, the <code>Event</code> object.
capturePhase	bool	When true, the handler is called during the capture phase of the event.

the capturing phase of the propagation. (default: **false**). Default is false, meaning the event is called in the bubbling phase if the target object is an ancestor of the target. Default is true for the at-target phase if this is the target.

void dispatchEvent ()

Simulates the occurrence of an event in this target. A script can create a **UIEvent** object for a specific event and pass it to this method to start the event propagation for the target.

void hide ()

Hides this element.

void notify (eventName:String)

Sends a notification message, simulating the specified interaction event.

Parameter	Type	Description
eventName	String	The name of the control event. One of: <code>onClick</code> , <code>onChange</code> , <code>onChange</code> . <code>onChange</code> simulates the <code>onChange</code> event for a control, an <code>onClick</code> event for a control that supports that event.

void onClick ()

An event-handler callback function, called when the element has been clicked.

void onShortcutKey ()

An event-handler callback function, called when the `shortcutKey` sequence is typed in the active window. In Windows only.

bool removeEventListener (eventName:String, handler:Function, capturePhase:bool=false)

Unregisters an event handler for a particular type of event in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles
capturePhase	bool	Whether to call the handl capturing phase of the ev (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children but when it is shown again, the children retain their states.

[Contents](#) :: [Index](#)

Class

ScriptUI

A global class containing central information about ScriptUI instantiable.

QuickLinks [getResourceText](#), [newFont](#), [newImage](#)

Properties

Property	Type	Access	Description
Alignment	String	readonly	Collects the enumerate that can be used in the <code>alignChildren</code> property of controls and containers. Predefined alignment values are TOP, BOTTOM, LEFT, RIGHT, FILL, CENTER.
FontStyle	Object	readonly	Collects the enumerate that can be used as the argument to the ScriptUI.newFont() function. Predefined styles are REGULAR, BOLD, ITALIC, BOLDITALIC.
applicationFonts	Object	readonly	The font constants defined for the host application.
compatibility	Object	readonly	An object whose properties are the names of compatibility modes supported by the host application. The presence of the <code>ScriptUI.compatibility.su1Pair</code> property means that the application is backward compatible with the coordinate system of F elements in ScriptUI v

coreVersion	String	readonly	A string containing the version number of the module.
frameworkName	String	readonly	A string containing the the UI component framework with which this version of ScriptUI is compatible
version	Any	readonly	A string containing the number of the ScriptUI component framework

Methods **String getResourceText (text:String)**

Finds and returns the resource for a given text string from the application's resource data.

If no string resource matches the given text, the text itself is

Parameter	Type	Description
text	String	The text to match.

ScriptUIFont newFont (name:String, style:String Number, size:Number)

Creates a new font object for use in text controls and titles.

Parameter	Type	Description
name	String	The font name, or the font family name
style	String Number	The font style; can be string, or one of the values of ScriptUIFont.Style.
size	Number	The font size in points.

ScriptUIImage newImage (normal:String, disabled:String, pressed:String, rollover:String)

Loads a new image from resources or disk files into an image object. Creates a new global image object for use in controls that can display images, loading the associated images from the specified resource names or image files.

Parameter	Type	Description
normal	String	The resource name or the disk file path

		image used for the normal state.
disabled	String	The resource name, or the disk file path image used for the disabled state.
pressed	String	The resource name, or the disk file path image used for the pressed state.
rollover	String	The resource name, or the disk file path image used for the rollover state.

[Contents](#) :: [Index](#)

Class

ScriptUIBrush

A painting brush that encapsulates a color or pattern. Create with [ScriptUIGraphics.newBrush\(\)](#). Use [backgroundColor](#) properties, and pass as an argument.

Properties

Property	Type	Access	Description
color	Array of Number	readonly	The brush color. The paint color SOLID_COLOR, A] specifying the color and the as numbers in the of 0 is fully transparent fully opaque.
theme	String	readonly	The theme name. The name of a theme when the type is colors are defined.
type	Number	readonly	The brush type. One of these constants: ScriptUIGraphics or ScriptUIGraphics

Element of [ScriptUIGraphics.backgroundColor](#)
[ScriptUIGraphics.disabledBackgroundColor](#)

Used in [ScriptUIGraphics.fillPath](#) (**brush**:*ScriptUIBrush*)

Return **ScriptUIBrush** [ScriptUIGraphics.newBrush](#) (**t**, *Number String*)

[Contents](#) :: [Index](#)

Class

ScriptUIFont

Encapsulates the qualities of a font used to draw text control.

Create with the [newFont\(\)](#) method. Used as a value Passed as an argument to [drawString\(\)](#) and [measureString\(\)](#)

Properties

Property	Type	Access	Description
family	String	readonly	The font family name
name	String	readonly	The complete font name consisting of the family name and style, if specified.
size	Number	readonly	The font point size
style	Object	readonly	The font style. On constants in ScriptUIFont
substitute	String	readonly	The name of a substitute font, a fallback font to substitute for this font if the requested font family is not available.

Element of [ScriptUIGraphics.font](#)

Used in [ScriptUIGraphics.drawString](#) ([text:String](#), [pen:ScriptUIFont](#), [x:Number](#), [y:Number](#), [font:ScriptUIFont](#))
[ScriptUIGraphics.measureString](#) ([text:String](#), [font:ScriptUIFont](#), [boundingWidth:Number](#))

Return [ScriptUIFont](#) [ScriptUI.newFont](#) ([name:String](#), [style:String](#), [size:Number](#))

[Contents](#) :: [Index](#)

Class

ScriptUIGraphics

An object used to draw custom graphics, window, container, and control objects.

Allows a script to customize aspects of the color and font. Use an *onDraw* callback to set properties or call the functions. All measurements are in pixels.

QuickLinks [closePath](#), [drawFocusRing](#), [drawImage](#), [ellipsePath](#), [fillPath](#), [lineTo](#), [measureString](#), [newPen](#), [onDraw](#), [rectPath](#), [strokePath](#)

Properties

Property	Type
BrushType	Object
PenType	Object
backgroundColor	ScriptUIBrush

currentPath [ScriptUIPath](#)

currentPoint [Point](#)

disabledBackgroundColor [ScriptUIBrush](#)

disabledForegroundColor [ScriptUIPen](#)

font [ScriptUIFont](#)

foregroundColor [ScriptUIPer](#)

Methods `void closePath ()`

Closes the current path.

Defines a line from the current position (the value of `currentPath`) to the current path (the value of `currentPath`).

`void drawFocusRing (left:Number, top:height:Number)`

Draws a focus ring within a region of the current path.

Parameter	Type	Description
left	Number	The left coordinate of the focus ring. Value is relative to the current path.
top	Number	The top coordinate of the focus ring. Value is relative to the current path.
width	Number	The width of the focus ring.
height	Number	The height of the focus ring.

`void drawImage (image:ScriptUIImage, width:Number, height:Number)`

Draws an image within a given region of the current path. Uses the version of the image that is appropriate for the current state.

Parameter	Type	Description
image	ScriptUIImage	The image to be drawn. This object must be an instance of <code>ScriptUIImage</code> and must be in a state that allows it to be drawn, such as <code>enabled</code> or <code>disabled</code> .

left	Number	The left coordinate of the origin
top	Number	The top coordinate of the origin
width	Number	The width of the control is stretched to the original width
height	Number	The height of the control is stretched to the original height

void drawOSControl ()

Draw the platform-specific control associated with the given control ID.

void drawString (text:String, pen:ScriptUIPen, font:ScriptUIFont)

Draw a string of text starting at a given point. The text is drawn using the given drawing pen and font.

Parameter	Type	Description
text	String	The text string to be drawn.
pen	ScriptUIPen	The drawing pen to use.
x	Number	The left coordinate of the upper-left corner of this element.
y	Number	The top coordinate of the upper-left corner of this element.
font	ScriptUIFont	The font to use for this object.

Point ellipsePath (left:Number, top:Number, width:Number, height:Number)

Defines an elliptical path within a given bounding box. The path is added to the `currentPath` object, which can be filled using `strokePath()`.

Returns a `Point` object for the upper-left corner of the bounding box. The `currentPoint` is updated to the new `currentPoint`.

Parameter	Type	Description
left	Number	The left coordinate origin of this element.
top	Number	The top coordinate origin of this element.
width	Number	The width of the element.
height	Number	The height of the element.

void fillPath (*brush:ScriptUIBrush*, *path:ScriptUIPath*)
 Fills a path using a given painting brush.

Parameter	Type	Description
brush	ScriptUIBrush	The brush used for painting.
path	ScriptUIPath	The path to be filled.

Point lineTo (*x:Number*, *y:Number*)
 Adds a path segment to the [currentPath](#).
 The line is defined from the [currentPoint](#) to the point (*x*, *y*).
 Returns the [Point](#) object for the destination value of [currentPoint](#).

Parameter	Type	Description
x	Number	The X coordinate relative to the origin.
y	Number	The Y coordinate relative to the origin.

Dimension measureString (*text:String*, *boundingWidth:Number*)
 Calculates the size needed to display a string in a given bounding width.
 Returns a [Dimension](#) object that contains the width and height of the string in pixels.

Parameter	Type	Description
text	String	The text to be measured.
font	ScriptUIFont	The font used for the text.

in this o

boundingWidth	Number	The bou
---------------	--------	---------

Point moveTo (*x:Number, y:Number*)

Adds a given point to the **currentPath**, at position.

Returns the **Point** object which is the new

Parameter	Type	Description
x	Number	The X coordinate origin of this element
y	Number	The Y coordinate origin of this element

ScriptUIBrush newBrush (*type:Number*)

Creates a new painting brush object.

Parameter	Type	Description
type	Number	The brush type, such as SOLID . One of the constants: ScriptUIGraphic , ScriptUIGraphic .
color	Array of Number String	The brush color. - If type is SOLID as an array of three values of the color (alpha channel). range [0.0..1.0]. 0 is transparent, and 1 is opaque. - If the type is TEXT of the theme. The color of the host application.

ScriptUIPath newPath ()

Creates a new, empty path object.

Replaces any existing path in **currentPath**.

ScriptUIPen newPen (*type*:Number, *color*: Array of Number String)

Creates a new drawing pen object.

Parameter	Type	Description
type	Number	The pen type, solid or theme. One of the constants <i>ScriptUIGraphics.PenType.SOLID_COLOR</i> or <i>ScriptUIGraphics.PenType.THEME_COLOR</i> .
color	Array of Number String	The pen color. - If type is <i>SOLID_COLOR</i> , the color expressed as an array of three or four values, in the form [R, B, G, A] specifying the red, green, and blue values of the color and, optionally, the opacity (alpha channel). All values are numbers in the range [0.0..1.0]. An opacity of 0 is fully transparent, and an opacity of 1 is fully opaque. - If the type is <i>THEME_COLOR</i> , the name string of the theme. Theme colors are defined by the host application.

void onDraw (*drawState*:DrawState)

A function that is called when the element needs to be drawn.

Parameter	Type	Description
drawState	DrawState	On object containing information about the current mouse button and modifier key states.

Point rectPath (*left*:Number, *top*:Number, *width*:Number, *height*:Number)

Defines a rectangular path in the *currentPath* object.

The rectangle can be filled using *fillPath()* or stroked using *strokePath()*. Returns the *Point* object for the upper left corner of the rectangle, which becomes the new value of *currentPoint*.

Parameter	Type	Description
left	Number	The left coordinate relative to the origin of this element.
top	Number	The top coordinate relative to the origin of this element.
width	Number	The width in pixels.
height	Number	The height in pixels.

void strokePath ([pen:ScriptUIPen](#), [path:ScriptUIPath](#))

Strokes the path segments of a path with a given drawing pen.

Parameter	Type	Description
pen	ScriptUIPen	The drawing pen that defines the color and line width.
path	ScriptUIPath	The path object. Default is the currentPath.

Element of

[Button.graphics](#)

[Checkbox.graphics](#)

[DropDownList.graphics](#)

[EditText.graphics](#)

[Group.graphics](#)

[IconButton.graphics](#)

[Image.graphics](#)

[ListBox.graphics](#)

[Panel.graphics](#)

[Progressbar.graphics](#)

[RadioButton.graphics](#)

[Scrollbar.graphics](#)

[Slider.graphics](#)

[StaticText.graphics](#)

[TreeView.graphics](#)

[Window.graphics](#)

[Contents :: Index](#)

Class

ScriptUIImage

Encapsulates a set of images that can be drawn into a control.

Different images can reflect the current state, such as a dimmed version for a disabled control. The `newImage()` method. Passed as an argument to `drawImage()`.

Properties

Property	Type	Access	Description
format	String	readonly	The image format. One of: resource, JPEG, GIF, TIFF, PNG, or PICT (Macintosh)
name	String	readonly	The image name. Either the file name, or the resource name.
pathname	String	readonly	The full path to the file that contains the image.
size	Dimension	readonly	The image size in pixels.

Element of [IconButton.image](#)
[Image.image](#)
[ListItem.image](#)

Used in [ScriptUIGraphics.drawImage](#)
(*image:ScriptUIImage*, *left:Number*,
top:Number, *width:Number*,
height:Number)

Return **ScriptUIImage** [ScriptUI.newImage](#)
(*normal:String*, *disabled:String*,
pressed:String, *rollover:String*)

[Contents](#) :: [Index](#)

Class

ScriptUIPath

A helper object that encapsulates a drawing path for a figure to be drawn into a window or control.

Create with the `newPath()`, `moveto()`, `lineto()`, `rectPath()`, and `ellipsePath()` methods. Used as value of `currentPath`, where it is acted upon by methods such as `closePath()`. Pass as optional argument to `fillPath()` and `strokePath()`, which otherwise act upon the current path.

Element of `ScriptUIGraphics.currentPath`

Used in `ScriptUIGraphics.strokePath (pen:ScriptUIPen, path:ScriptUIPath)`
`ScriptUIGraphics.fillPath (brush:ScriptUIBrush, path:ScriptUIPath)`

Return **ScriptUIPath** `ScriptUIGraphics.newPath ()`

[Contents](#) :: [Index](#)

Class

ScriptUIPen

A drawing pen that defines a color and line width us
Create with [ScriptUIGraphics.newPen\(\)](#). Use as a va
[foregroundColor](#) properties, and pass as an argumen
to [drawString\(\)](#) and [strokePath\(\)](#) methods.

Properties

Property	Type	Access	Description
color	Array of Number	readonly	The pen color. The paint color to u SOLID_COLOR. A G, A] specifying the of the color and the value as numbers in opacity of 0 is fully opacity of 1 is fully
lineWidth	Number	r/w	The pixel width of t
theme	String	readonly	The theme name. The name of a color when the type is TH colors are defined b
type	String	readonly	The pen type, solid One of these consta ScriptUIGraphics.P or ScriptUIGraphics.P

Element of

[ScriptUIGraphics.disabledForegroundColor](#)
[ScriptUIGraphics.foregroundColor](#)

Used in

[ScriptUIGraphics.strokePath](#) ([pen:ScriptUIPen](#), [path](#)
[ScriptUIGraphics.drawString](#) ([text:String](#), [pen:Scrip](#)
[y:Number](#), [font:ScriptUIFont](#))

Return

ScriptUIPen [ScriptUIGraphics.newPen](#) ([type:Numb](#)

Number String)

[Contents :: Index](#)

Class

Scrollbar

A scrollbar with a draggable scroll indicator and stepper buttons to move the indicator.

The scrollbar control has a horizontal orientation if the width is greater than the height at creation time, or vertical if its height is greater than its width. Calls the `onChange` callback after the position of the indicator is changed or if its `notify()` method is called. Calls the `onChange` callback repeatedly while the user is moving the indicator. Scrollbars are often created with an associated `EditText` field to display the current value of the scrollbar, and to allow setting the scrollbar's position to a specific value.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onChange](#), [onChangeing](#), [onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	True if this scrollbar is active. An active scrollbar is the one with keyboard focus, that is, the one that accepts key events or in the case of a Button, is shown when the user presses Return or Enter. In Windows, the scrollbar has focus when the space bar is pressed in OS.
<code>alignment</code>	<code>String</code>	r/w	The alignment for this element. If not defined, this property overrides the <code>alignChildren</code> property.

setting for t container.

This can be string, which indicates the alignment for orientation in the parent container, or an array of two indicating the horizontal and vertical alignment (in that order). Allowed values depend on the orientation of the parent container. They are not case sensitive.

- For orientation="bottom", fill

- For orientation="left, right", fill

- For orientation="bottom, left, right", fill

bounds

Bounds

r/w

The bounds of the element are defined by the parent-relative coordinates.

Setting an element's size or location changes its position property, and

			versa.
children	Array of Object	readonly	An array of elements.
enabled	bool	r/w	<p>True if this is enabled.</p> <p>An enabled can accept i according to When false elements do accept input types of ele have a dimi appearance</p>
graphics	ScriptUIGraphics	readonly	The graphic that can be customize t element's appearance response to onDraw even
helpTip	String	r/w	The help te displayed w mouse hove the element
indent	Number	r/w	<p>The numbe pixels to in element du automatic l.</p> <p>Applies for orientation alignment, orientation alignment.</p>

jumpdelta	Number	r/w	<p>The amount of increment or decrement of scrollbar in position when user clicks on behind the scrollbar element.</p> <p>Default is 2 pixels in the range between the maximum and minimum pixel values.</p>
location	Point	r/w	<p>The upper left corner of the scrollbar element relative to its parent.</p> <p>The location is defined as [bounds.x, bounds.y]. When an element changes its position, the property, and vice versa.</p>
maximumSize	Dimension	r/w	<p>The maximum height and width at which the scrollbar can be resized.</p>
maxvalue	Number	r/w	<p>The maximum value allowed in the value property.</p> <p>Together with minvalue, sets the scrolling range. Default is 100.</p>

minimumSize	Dimension	r/w	The minimum height and width of the element when it is resized.
minvalue	Number	r/w	The minimum value allowed in the value property. Together with maxvalue, sets the scrolling range. Default is 0.
parent	Object	readonly	The parent element.
preferredSize	Dimension	r/w	The preferred size used by layout managers to determine the size for each element. If not explicitly set by a script, the established UI framework which Script uses, or the browser, based on the style attributes of the element as well as font, font size, and other framework-specific attributes. A script can explicitly set preferredSize when the layout manager is invoked in order to establish

			element size than the default
properties	Object	r/w	<p>An object that contains one or more properties. The object is created at the time the scrollbar is created (properties are added when the scrollbar is created).</p> <p>A Scrollbar has no creation properties. The add() method creates it with an initial value of 0. The first two arguments are the minimum and maximum values of the range.</p>
shortcutKey	String	r/w	<p>The key sequence that invokes the onShortcutKey event for this element. The default is Windows o.</p>
size	Dimension	r/w	<p>The current dimensions of the scrollbar element.</p> <p>Initially undefined and unless overridden, is set by a scrollbar script defined by the LayoutManager. A scrollbar script can easily set the size before the scrollbar is created.</p>

layout manager is invoked to lay out an element other than the scrollbar. The default size is not recomputed. Defined as [bounds.width, bounds.height]. Setting an element's size changes the scrollbar's position vice-versa.

stepdelta	Number	r/w	The amount by which to increase or decrease the scrollbar element's position when the user clicks the scrollbar button.
-----------	--------	-----	---

type	String	readonly	The element's type, "scrollbar".
------	--------	----------	----------------------------------

value	Number	r/w	The current value of the indicator. If set to a value outside the range specified by minvalue and maxvalue, it is automatically set to the closest boundary.
-------	--------	-----	--

visible	bool	r/w	True if this scrollbar is shown, false if it is hidden.
---------	------	-----	---

When a container is hidden, its children are also hidden. However, if the children have the visibility variable set to `visible`, they retain their visibility and are shown accordingly. If the parent is shown, the children are also shown.

window	Window	readonly	The window of this element.
windowBounds	Bounds	readonly	The bounds of the element relative to the top-level window.

Methods `bool addEventListener (eventName:String, handler:Function, capturePhase:bool=false)`

Registers an event handler for a particular type of event occurring on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaFX Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be executed when the event occurs. A handler takes one argument, the <code>UIEvent</code> .
capturePhase	bool	When true, the handler is called during the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if

object is an ancestor of the target
the at-target phase if this object is
the target.

Event **dispatchEvent** ()

Simulates the occurrence of an event in this target.

A script can create a **UIEvent** object for a specific event and pass it to this method to start the event propagation for the event.

void hide ()

Hides this element.

void notify (*eventName:String*)

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . By default, this method simulates the <code>onChange</code> event for an edit control, an <code>onClick</code> event for controls that do not support that event.

void onChange ()

An event-handler callback function, called when the user has finished dragging the position indicator, or has clicked the position indicator.

void onChanging ()

An event-handler callback function, called when the control element is in the process of changing its position.

The handler is called for any motion of the position indicator while this control has the input focus.

void onShortcutKey ()

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window.

In Windows only.

bool removeEventListener (*eventName:String, handler:Function, capturePhase:bool=false*)

Unregisters an event handler for a particular type of event (specified by `eventName`) in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only during the capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden, but when it is shown again, the children retain their own visibility states.

Class

Slider

A slider bar that indicates a numeric value with a moveable position indicator.

All slider controls have a horizontal orientation. Calls the `onChange` callback after the position of the indicator is changed or if its `notify` method is called. Calls the `onChangeing` callback repeatedly while the user is moving the indicator. The `value` property contains the current position of the indicator within the range of `minvalue` and `maxvalue`.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onChange](#), [onChangeing](#), [onShortcutKey](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	True if this element is active. An active control is the one with keyboard focus—that is, the one that accepts keystrokes or in the case of a Button, is selected when the user types Return or Enter in Windows, or the space bar in MacOS.
<code>alignment</code>	<code>String</code>	r/w	The alignment style for this element. If defined, this value overrides the <i>alignChildren</i> setting for the parent container. This can be a single

string, which indicates the alignment for the orientation specified in the parent container, or an array of two strings indicating both the horizontal and vertical alignment (in that order). Allowed values depend on the orientation value the parent container has. They are not case sensitive.

- For orientation=row: top, bottom, fill
- For orientation=column: left, right, fill
- For orientation=stack: bottom, left, right, fill

bounds	Bounds	r/w	The boundaries of the element, in parent-relative coordinates. Setting an element's size or location changes its bounds property, and vice versa.
--------	---------------	-----	--

children	Array of Object	readonly	An array of child elements.
----------	-----------------	----------	-----------------------------

enabled	bool	r/w	<p>True if this element is enabled.</p> <p>An enabled element can accept input, according to its type. When false, control elements do not accept input, and types of elements have a dimmed appearance.</p>
graphics	ScriptUIGraphics	readonly	<p>The graphics object that can be used to customize the element's appearance, in response to the <code>onDraw</code> event.</p>
helpTip	String	r/w	<p>The help text that is displayed when the mouse hovers over the element.</p>
indent	Number	r/w	<p>The number of pixels to indent the element during automatic layout.</p> <p>Applies for column orientation and left alignment, or row orientation and top alignment.</p>
location	Point	r/w	<p>The upper left corner of this element relative to its parent.</p>

			The location is defined as [bounds.x, bounds.y]. Setting an element's location changes its bounding property, and vice versa.
maximumSize	Dimension	r/w	The maximum height and width which the element can be resized.
maxvalue	Number	r/w	The maximum value allowed in the value property. Together with minvalue, sets the range. Default is 1
minimumSize	Dimension	r/w	The minimum height and width to which the element can be resized.
minvalue	Number	r/w	The minimum value allowed in the value property. Together with maxvalue, sets the range. Default is 0
parent	Object	readonly	The parent element
preferredSize	Dimension	r/w	The preferred size used by layout managers to determine the best size for each

element.

If not explicitly set by a script, value established by the UI framework in which ScriptUI is employed, and is based on such attributes of the element as its text font, font size, icon size, and other UI framework-specific attributes. A script can explicitly set preferredSize before the layout manager is invoked in order to establish an element size other than the default.

properties

Object

r/w

An object that contains one or more creation properties of the container (properties used when the element is created).

A Slider object has no creation properties. The third argument of the add() method that creates it is the initial value, and the fourth and fifth arguments are the

			minimum and maximum values the range.
shortcutKey	String	r/w	The key sequence that invokes the <code>onShortcutKey</code> callback for this element (Windows only).
size	Dimension	r/w	<p>The current dimensions of this element.</p> <p>Initially undefined and unless explicitly set by a script, it is defined by a <code>LayoutManager</code>. A script can explicitly set size before the layout manager is invoked to establish an element size other than the <i>preferredSize</i> or the default size, but this is not recommended. Defined as <code>[bounds.width, bounds.height]</code>. Setting an element size changes its bounds property, vice-versa.</p>
type	String	readonly	The element type "slider".
value	Number	r/w	The current position of the indicator.

				If set to a value outside the range specified by minvalue and maxvalue, it is automatically reset to the closest boundary.
visible	bool	r/w		True if this element is shown, false if it is hidden. When a container is hidden, its children are also hidden, but they retain their own visibility values, and are shown or hidden accordingly when the parent is next shown.
window	Window	readonly		The window that this element belongs to.
windowBounds	Bounds	readonly		The bounds of this element relative to the top-level parent window.

Methods **bool addEventListener** (*eventName:String*, *handler:Function* [, *capturePhase:bool=false*])

Registers an event handler for a particular type of event occurring on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .

handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be executed when the event occurs. A handler function takes one argument, the UIEvent object.
capturePhase	bool	When true, the handler is called only in the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself the target.

Event **dispatchEvent** ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass to this method to start the event propagation for the event.

void **hide** ()

Hides this element.

void **notify** (*eventName:String*)

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler to call. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . By default, <code>onChange</code> event for an edittext control, an <code>onClick</code> event for controls that support that event.

void **onChange** ()

An event-handler callback function, called when the user has finished dragging the position indicator, or has clicked the control.

void **onChanging** ()

An event-handler callback function, called when the content of the

element is in the process of changing
The handler is called for any motion of the position indicator when this control has the input focus.

void onShortcutKey ()

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window.

In Windows only.

bool removeEventListener (eventName:String, handler:Function capturePhase:bool=false)

Unregisters an event handler for a particular type of event occurring in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only in the capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visibility states.

Class

StaticText

A text label that the user cannot change.

QuickLinks [addEventListener](#), [dispatchEvent](#), [hide](#), [notify](#), [onShortcut](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
active	bool	r/w	Always false. An active element can have input focus. An active element is the one with keyboard focus, that is, the element that accepts keyboard input. In the context of a Button, isFocus is true when the button has the focus. Return key (Enter) on Windows, space bar on Mac OS.
alignment	String	r/w	The alignment for this element. If not defined, the element inherits the alignment setting for its container. This can be a string, which indicates the alignment orientation in the parent container.

container, array of tv indicating horizontal vertical al (in that or Allowed v depend or orientation the parent They are r sensitive.
 - For orientation bottom, fill
 - For orientation left, right, fil
 - For orientation bottom, left,

bounds	Bounds	r/w	The bound the eleme parent-rel. coordinate Setting an size or loc changes it property, i versa.
characters	Number	r/w	A number characters to reserve when calc preferred element.
children	Array of Object	readonly	An array c

			elements.
enabled	Bool	r/w	True if this is enabled An enable can accept according When false elements (accept input types of elements have a default appearance
graphics	ScriptUIGraphics	readonly	The graphics that can be customized element's appearance response to onDraw event
helpTip	String	r/w	The help text displayed mouse over the element
indent	Number	r/w	The number of pixels to indent element default is automatic Applies for orientation alignment orientation alignment
justify	String	r/w	The text justification style. (default is left) One of left

			right. Justif only work value is se creation o element.
location	Point	r/w	The upper corner of 1 element re its parent. The locati defined as [bounds.x bounds.y] an elemen changes it property, & versa.
maximumSize	Dimension	r/w	The maxim height and which the can be res
minimumSize	Dimension	r/w	The minir and width the elemen resized.
parent	Object	readonly	The paren
preferredSize	Dimension	r/w	The prefer used by la managers determine size for ea element. If not expi by a scrip establishe

UI framework which Screen employed based on s attributes element a font, font size, and c framework attributes. can explicit preferredS the layout is invoked to establish element si than the d

properties

Object

r/w

An object contains c creation p of the con (propertie when the created).

Creation p of a Static object can - multiline false (the the contro a single li When true control di: multiple l which cas wraps wit width of t - scrolling

			false (the the display cannot be When true displayed vertically using the and Down this case i multiline=
shortcutKey	String	r/w	The key s that invoc onShortcutKe for this el Windows
size	Dimension	r/w	The curre dimension element. Initially u and unless set by a sc defined by LayoutMa script can set size be layout ma invoked to an elemen other than <i>preferredS</i> default siz is not rec Defined a [bounds.w bounds.he Setting an size chang

			bounds pr vice-versa
text	String	r/w	The text to localizabl
type	String	readonly	The eleme "statictext
visible	bool	r/w	True if thi is shown, is hidden. When a co hidden, its are also hi they retain visibility v are shown according the parent shown.
window	Window	readonly	The windo this eleme to.
windowBounds	Bounds	readonly	The bound element re the top-lev window.

Methods **bool addEventListener** (*eventName:String*, *handler:Fun*
capturePhase:bool=false)
Registers an event handler for a particular type of event o
this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>Jc Tools Guide</i> .

handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be called when the event occurs. A handler takes one argument, the <code>UIEvent</code> .
capturePhase	bool	When true, the handler is called during the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if the target object is an ancestor of the target object. The handler is not called during the at-target phase if this object is the target.

Event dispatchEvent ()

Simulates the occurrence of an event in this target. A script can create a `UIEvent` object for a specific event and pass it to this method to start the event propagation for the event.

void hide ()

Hides this element.

void notify (eventName:String)

Sends a notification message, simulating the specified user interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handler. One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . <code>onChange</code> simulates the <code>onChange</code> event for an element control, an <code>onClick</code> event for controls that support that event.

void onShortcutKey ()

An event-handler callback function, called when the element's `shortcutKey` sequence is typed in the active window. In Windows only.

bool removeEventListener (eventName:String, handler:..

`capturePhase:bool=false])`

Unregisters an event handler for a particular type of event in this element.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event.
capturePhase	bool	Whether to call the handler only during the capturing phase of the event propagation (default: false)

`void show ()`

Shows this element.

When a window or container is hidden, its children are also hidden, but when it is shown again, the children retain their own visible states.

Class

TreeView

A hierarchical list whose items can contain child items.

The [ListItem](#) children of this control (in the `items` array) can *node*, which means that they can contain child items. An individual item can be expanded, so that the child items are displayed, or collapsed, so that the child items are hidden. Individual items can be selected at any level of the tree.

QuickLinks [add](#), [addEventListener](#), [dispatchEvent](#), [find](#), [hide](#), [notify](#), [removeAll](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
<code>active</code>	<code>bool</code>	r/w	<p>True if this control is active.</p> <p>An active control is the one with keyboard focus. That is, the control that accepts keyboard input. For example, in the case of a <code>Button</code>, it is set to <code>true</code> when the user presses the Return or Enter key in Windows, or the space bar in OS X.</p>
<code>alignment</code>	<code>String</code>	r/w	<p>The alignment for this element. If defined, this overrides the <code>alignChildren</code> setting for the container.</p> <p>This can be any string, which</p>

indicates the alignment of the element in the parent container, consisting of an array of two values indicating the horizontal and vertical alignment (in that order). Allowed values depend on the orientation of the parent container. They are not case sensitive.

- For orientation: top, bottom, fill
- For orientation: left, right, fill
- For orientation: top, bottom, left, right, fill

bounds	Bounds	r/w	The bounds of the element relative to its parent-related coordinates. Setting an element's size or location changes its property, and vice versa.
children	Array of Object	readonly	An array of elements.
enabled	bool	r/w	True if this

is enabled.

An enabled can accept according t When false elements do accept input types of ele have a dim. appearance

graphics	ScriptUIGraphics	readonly	The graphics that can be customize t element's appearance response to onDraw even
----------	----------------------------------	----------	---

helpTip	String	r/w	The help te displayed v mouse hov the element
---------	--------	-----	---

indent	Number	r/w	The numbe pixels to in element du automatic l Applies for orientation alignment, orientation alignment.
--------	--------	-----	--

itemSize	Dimension	r/w	The width : height in pi each item i Used by au to determin
----------	---------------------------	-----	--

			preferredSize (if not otherwise specified. If explicitly, then each item must match the length and height and width among all items in the list
items	Array of ListItem	readonly	<p>The array of level choices displayed in the list.</p> <p>Access this property with a 0-based index. To obtain the number of items in the list, use <code>items.length</code>. The objects are <code>ListItem</code> objects when items are specified on the parent object, or a <code>ListItem</code> object, or a <code>ListItem</code> object using the list control's <code>addItem</code> method.</p>
location	Point	r/w	<p>The upper-left corner of the element relative to its parent.</p> <p>The location is defined as <code>[bounds.x, bounds.y]</code>. When an element changes its location, its <code>location</code> property, and</p>

			versa.
maximumSize	Dimension	r/w	The maximum height and width of the element which the element can be resized.
minimumSize	Dimension	r/w	The minimum height and width of the element which the element can be resized.
parent	Object	readonly	The parent of the element.
preferredSize	Dimension	r/w	The preferred size of the element used by layout managers to determine the size for each element. If not explicitly set by a script, the size is established by the UI framework which Scrip employed, based on suitable attributes of the element as font, font size, and other framework attributes. An element can explicitly set its preferredSize attribute to influence the layout manager when it is invoked to establish the size of the element, which is larger than the default size.

properties	Object	r/w	<p>An object that contains on creation properties of the control (properties when the element is created).</p> <p>Creation properties of a ListBox can include</p> <ul style="list-style-type: none"> - <i>items</i>: An array of strings for each top-level item. An item is created for each item. An item's text string creates a selection item. Supplemental property, or <i>items</i> argument to the add() method, not both. This is most useful for elements defined using Resource Specifications.
selection	ListItem	r/w	<p>The currently selected list item.</p> <p>Setting this property causes the selected item to be highlighted and scrolled into view if necessary. If no items are selected, the value is null.</p>

to null to de
 items. You c
 the value us
 index of an
 rather than
 reference. I
 an index va
 is out of ran
 operation is
 When set w
 index value
 property sti
 an object re

shortcutKey	String	r/w	The key se that invoke onShortcutKey for this ele Windows o
-------------	--------	-----	---

size	Dimension	r/w	The current dimensions element. Initially un and unless set by a scr defined by LayoutMar script can e set size bef layout man invoked to an element other than t <i>preferredSi</i> default size is not recor Defined as [bounds.wi
------	---------------------------	-----	--

			bounds.height; Setting an element's size change bounds provides vice-versa.
type	String	readonly	The element's type, the string "treeview".
visible	bool	r/w	True if this element is shown, false if it is hidden. When a container is hidden, its children are also hidden; they retain their visibility value and are shown or hidden accordingly when the parent is shown.
window	Window	readonly	The window containing this element.
windowBounds	Bounds	readonly	The bounds of the element relative to the top-level window.

Methods [ListItem](#) **add** (*type:String*, *text:String*)
 Adds an item to the top-level choices in this list.
 Returns the item control object.

Parameter	Type	Description
type	String	The type of the child element, the string "item".
text	String	The localizable text label for the item.

bool addEventListener (*eventName:String*, *handler:Function*, *capturePhase:bool=false*)

Registers an event handler for a particular type of event on this element.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>Java Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be executed when the event occurs. A handler takes one argument, the <code>UIEvent</code> object.
capturePhase	bool	When true, the handler is called during the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if the target object is an ancestor of the target object, or the at-target phase if this object is the target.

Event dispatchEvent ()

Simulates the occurrence of an event in this target.

A script can create a `UIEvent` object for a specific event and pass it to this method to start the event propagation for the event.

ListItem find (*text:String*)

Retrieves an item object from the list that has a given text label.

Parameter	Type	Description
text	String	The text string to match.

void hide ()

Hides this element.

void notify (*eventName:String*)

Sends a notification message, simulating the specified user.

interaction event.

Parameter	Type	Description
eventName	String	The name of the control event handle One of: <code>onClick</code> , <code>onChange</code> , <code>onChanging</code> . By simulates the <code>onChange</code> event for an edit control, an <code>onClick</code> event for controls that support that event.

void remove (*what:Any*)

Removes a child item from the list.

Parameter	Type	Description
what	Any	The item or child to remove, specified based index in the top-level item list, its value, or as a <code>ListItem</code> object.

void removeAll ()

Removes all child items from the list.

bool removeEventListener (*eventName:String, handler:Function, capturePhase:bool=false*)

Unregisters an event handler for a particular type of event in this element.

All arguments must be identical to those that were used to the event handler.

Parameter	Type	Description
eventName	String	The name of the event.
handler	Function	The function that handles the event
capturePhase	bool	Whether to call the handler only capturing phase of the event propagation (default: false)

void show ()

Shows this element.

When a window or container is hidden, its children are also hidden but when it is shown again, the children retain their own visual states.

Contents :: Index

Class

UIEvent

Encapsulates input event information for an event that propagates through a container and control hierarchy.

Implements W3C standard event handling. This object is passed to a function that you register to respond to events of a certain type that occur in a window or control. Use `windowObj.addEventListener()` or `controlObj.addEventListener()` to register a handler function.

QuickLinks [UIEvent](#), [initEvent](#), [initUIEvent](#), [preventDefault](#), [stopPropagation](#)

Properties

Property	Type	Access	Description
bubbles	bool	readonly	True if the event is of a type that bubbles.
cancelable	bool	readonly	True if the default action associated with the event can be canceled with <code>preventDefault()</code> .
captures	bool	readonly	True if this event can be captured.
currentTarget	bool	readonly	The event target object which is currently handling the event. During capturing and bubbling, this is different from the property target.
detail	Any	readonly	Passed in to <code>initEvent()</code> ; it can e.g. be a mouse click count.

eventPhase	String	readonly	The current phase of event propagation; one of none, target, capture, bubble.
target	Object	readonly	The event target object for this event.
timeStamp	Date	readonly	The date and time at which the event occurred.
type	String	readonly	The name of the event that this Event object represents.
view	Any	readonly	

Methods **UIEvent** **UIEvent** (*type:String* [, *captures:bool=false*] [, *bubbles:bool=false*], *view:Object*, *detail:Number*)

Creates an event.

The UIEvent object is normally created by ScriptUI and passed to your event handler. However, you can simulate a user action by constructing an event object and sending it to a target object's `dispatchEvent()` function.

Parameter	Type	Description
type	String	The event type. See <code>UIEvent.type</code> property.
captures	bool	Set to true if this event can be captured. (default: false)
bubbles	bool	Set to true if the event bubbles. (default: false)
view	Object	The ScriptUI element that this event relates to.
detail	Number	This value sets the detail property; it can e.g. be a mouse click count.

void **initEvent** (*type:String* [, *captures:bool=false*] [,

bubbles:bool=false] [, **cancelable:bool=false**])

Initializes an UI event as a core W3C event.

Parameter	Type	Description
type	String	The event type.
captures	bool	Set to true if this event can be captured. (default: false)
bubbles	bool	Set to true if the event bubbles. (default: false)
cancelable	bool	Set to true if the default action is cancelable. (default: false)

void initUIEvent (**type:String** [, **captures:bool=false**] [, **bubbles:bool=false**], **view:Object**, **detail:Number**)

Initializes an event.

Parameter	Type	Description
type	String	The event type.
captures	bool	Set to true if this event can be captured. (default: false)
bubbles	bool	Set to true if the event bubbles. (default: false)
view	Object	The ScriptUI element that this event relates to.
detail	Number	This value sets the detail property; it can e.g. be a mouse click count.

void preventDefault ()

Calling this function causes the default action associated with this event not to be called.

void stopPropagation ()

Stops the propagation of this event.

Return **UIEvent** **UIEvent.UIEvent** (**type:String** [, **captures:bool=false**] [, **bubbles:bool=false**],

`view:Object, detail:Number)`
UIEvent `Window.dispatchEvent ()`

[Contents :: Index](#)

Class

Window

The instance represents a top-level window or dialog box, with its child elements.


The globally available [Window](#) object provides access to pre-defined windows.

QuickLinks [Window](#), [add](#), [addEventListener](#), [alert](#), [center](#), [close](#), [confirm](#), [notify](#), [onClose](#), [onDraw](#), [onMove](#), [onMoving](#), [onResize](#), [onShow](#), [prompt](#), [remove](#), [removeEventListener](#), [show](#)

Properties

Property	Type	Access	Description
active	bool	r/w	Set to true to activate the window. <ul style="list-style-type: none">- A modal dialog box definition that is active.- An active window.- An active window with focus—that is, it receives keystrokes, mouse clicks, or other input events, or is selected via the Tab or Enter key.
alignChildren	String	r/w	Tells the layout manager how the children should be aligned within the container. Order of creation of children are preserved. Children created to the left of a container are aligned to the left of its container. Children created to the right of its container are aligned to the right of its container. See the <code>alignChildren</code> property for more details on the values.

alignment	String	r/w	<p>The alignment of a container overrides the the parent container.</p> <p>This can be indicates the orientation of container, or indicating by vertical alignment. Allowed values orientation of container. The</p> <ul style="list-style-type: none"> - For orientation - For orientation - For orientation <p>right, fill</p>
bounds	Bounds	r/w	The bounds area, excluding coordinates.
cancelElement	Object	r/w	<p>For window element to n presses a combination</p> <p>The cancella By default, l name or text disregarded)</p>
characters	Number	r/w	A number of reserve space preferred size
children	Array of Object	readonly	<p>The collection have been a</p> <p>An array including string contain</p>

			The length p number of c elements, an
defaultElement	Object	r/w	For window element to n presses a En By default, l name or text disregarded)
enabled	bool	r/w	True if this e An enabled according to control elem and all types dimmed app
frameBounds	Bounds	readonly	The bounds screen coord The frame c borders that of a window windowing :
frameLocation	Point	r/w	The top left frame in scr The same as frameBound move the wi specified loc The frameB accordingly.
frameSize	Dimension	readonly	The size and frame in scr
frameworkName	String	readonly	 Deprecate Use ScriptU

graphics	ScriptUIGraphics	readonly	The graphics customize the appearance of the container in response to the mouse.
helpTip	String	r/w	The help text displayed when the mouse hovers over the container.
indent	Number	r/w	The number of pixels to indent the container's child elements.
justify	String	r/w	The default justification for the container's child text elements. One of left, center, right, or justified. Only works in the context of text creation of the container.
layout	LayoutManager	r/w	The layout manager used to manage the container's child elements. The first time the container is made visible, the layout manager's layout function is called. Default is LayoutManager , which automatically manages the container's child elements.
location	Point	r/w	The upper-left corner of the container's drawable area. The same as the location property of the container's graphics property.
margins	Number	r/w	The number of pixels to margin the container's child elements. You can specify a value for each edge of the container. The default value is based on the platform and is chosen by the Adobe UI guidelines.
maximized	bool	r/w	True if the container is maximized.

maximumSize	Dimension	r/w	The largest size a window can be.
minimized	bool	r/w	True if the window is iconified.
minimumSize	Dimension	r/w	The smallest size a window can be.
orientation	String	r/w	The layout orientation of the container. Interpreted by the container's LayoutManager (case-insensitive). For window, the default is horizontal. For JPanel, the default is vertical. For JApplet, the default is vertical. For JComponent, the default is the container's orientation.
parent	Object	readonly	The immediate parent of this window.
preferredSize	Dimension	r/w	The preferred size of the window. Used in automatic layout.
properties	Object	r/w	An object that contains the creation properties of the window (properties used by the container to create the window). Creation properties of the window object can include: - resizable: true if the window can be resized, false otherwise. - title: the title of the window. - su1PanelC: true if the window is a subwindow of a JComponent only. When true, the window's position and size are determined by the positions of the parent window and the component.

compatibility which the value is measured from the top-left corner. Default is false. To override the default, use the `- closeButton` property on the title bar of the window, or the `closeButton` property of the `WindowType` enum. Does not apply to dialog boxes.

- `maximizeEnabled`: If true, the title bar contains the maximize button. If false, the title bar does not contain the maximize button. The default is true for typical desktop windows and false for typical mobile windows. No effect on dialog boxes.

shortcutKey	String	r/w	The keypress invokes this callback.
size	Dimension	r/w	The current content area coordinates.
spacing	Number	r/w	The number of pixels between child elements. Because each row of a single spaced container is a single spaced container. Therefore, the spacing depends on the type of container. To match standard guidelines, use 16 pixels for desktop and 8 pixels for mobile.
text	String	r/w	The title, label, or subtitle of the window.

			localizable
			Does not apply to a group.
type	String	readonly	The element type or "window"
version	Any	readonly	> Deprecate Use ScriptUI
visible	bool	r/w	When true, the window is visible when false it is hidden. When a container's children are shown or hidden, the children retain their visibility if the parent is visible or hidden.
window	Window	readonly	The window to which the child belongs.
windowBounds	Bounds	readonly	The bounds of the top-level window.

Methods **void Window** (*type:String*, *title:String*, *bounds:Bounds*, *prop:Property*)
Creates a new window.

Parameter	Type	Description
type	String	The window type. One of: - <i>window</i> : Creates a simple window that can be used as a window for an application. (Not supported by Photoshop CS3.) - <i>palette</i> : Creates a modeless dialog, also known as a palette. (Not supported by Photoshop CS3.) - <i>dialog</i> : Creates a modal dialog. This argument can also be a ScriptUI reference. In that case, all other arguments are ignored.

title String

bounds

Bounds

properties

Object

Object **add** (*type:String*, *bounds:Bounds*, *text:String*, *properties:Object*)

Creates and returns a new control or container object and adds it to the children of this window.

Parameter	Type	Description
-----------	------	-------------

type	String	The type of the child element, as specified for the type property. Control types are listed in the <i>JavaScript Tools Guide</i> .
bounds	Bounds	A bounds specification that describes the size and position of the new control or container, relative to its parent. If supplied, this value creates a new Bounds object which is assigned to the new object's <code>bounds</code> property.
text	String	The text or label, a localizable string. Initial text to be displayed in the control as the title, label, or contents, depending on the control type. If supplied, this value is assigned to the new object's <code>text</code> property.
properties	Object	An object that contains one or more creation properties of the new child (properties used only when the element is created). The creation properties depend on the element type. See <code>properties</code> property of each control type.

bool addEventListener ([eventName:String](#), [handler:Function](#) [, [capturePhase:bool=false](#)])

Registers an event handler for a particular type of event occurring in this window.

Parameter	Type	Description
eventName	String	The name of the event. Event names are listed in the <i>JavaScript Tools Guide</i> .
handler	Function	The function that handles the event. This can be the name of a function defined in the extension, or a locally defined handler function to be executed when the event occurs. A handler function takes one argument, the UIEvent object.
capturePhase	bool	When true, the handler is called only in the capturing phase of the event propagation. (default: false) Default is false, meaning that the handler is called in the bubbling phase if this object is an ancestor of the target, or in the at-target phase if this object is itself

the target.

void alert (*message:String*, *title:String* [, *errorIcon:bool=false*])

Displays a platform-standard dialog containing a short message and an OK button.

Parameter	Type	Description
message	String	The string for the displayed message.
title	String	A string to appear as the title of the dialog, if the platform supports a title. Ignored in Mac OS, which does not support titles for alert dialogs. The default title string is "Script Alert".
errorIcon	bool	When true, the platform-standard alert icon is replaced by the platform-standard error icon in the dialog. (default: false) Ignored in Mac OS, which does not support icons for alert dialogs.

void center (*window:Window*)

Centers this window on screen or with respect to another window.

Parameter	Type	Description
window	Window	The relative window. If not specified, centers on the screen.

void close (*return:Any*)

Closes this window.

. If an `onClose` callback is defined for the window, calls that function before closing the window.

Parameter	Type	Description
return	Any	A number to be returned from the <code>show()</code> method that invoked this window as a modal dialog.

bool confirm (*message:String* [, *noAsDefault:bool=false*], *title:String*)

Displays a platform-standard dialog containing a short message and two buttons labeled Yes and No.

Returns true if the user clicked Yes, false if the user clicked No.

Parameter	Type	Description
-----------	------	-------------

message	String	The string for the displayed message.
noAsDefault	bool	When true, the No button is the default choice, selected when the user types Enter. (default: false) Default is false, meaning that Yes is the default choice.
title	String	A string to appear as the title of the dialog, if the platform supports a title. Ignored in Mac OS, which does not support titles for alert dialogs. The default title string is "Script Alert".

UIEvent dispatchEvent ()

Simulates the occurrence of an event in this target.

A script can create a [UIEvent](#) object for a specific event and pass it to this method to start the event propagation for the event.

Window find (type:String, title:String)

Use this method to find an existing window.

This includes windows defined by ScriptUI resource strings, windows already created by a script, and windows created by the application (if the application supports this case). This function is not supported by all applications. Returns a [Window](#) object found or generated from the resource, or null if no such window or resource exists.

Parameter	Type	Description
type	String	The name of a predefined resource available to JavaScript in the current application; or the window type. If a title is specified, the type is used if more than one window with that title is found. Can be null or the empty string.
title	String	The window title.

void hide ()

Hides this windows.

When a window is hidden, its children are also hidden, but when it is shown again, the children retain their own visibility states.

- For a modal dialog, closes the dialog and sets its result to 0.

void notify (eventName:String)

Sends a notification message to all listeners, simulating the specified user

interaction event.

Parameter	Type	Description
eventName	String	The event name; if omitted, the default event is sent. One of: onClose , onMove , onMoving , onResize , onResizing , onShow

bool onClose ()

An event-handler callback function, called when the window is about to be closed.

Called when a request is made to close the window, either by an explicit call to the `close()` function or by a user action (clicking the OS-specific close icon in the title bar). The function is called before the window actually closes; it can return `false` to cancel the close operation.

void onDraw ()

An event-handler callback function, called when the window is about to be drawn.

Allows the script to modify or control the appearance, using the control's associated [ScriptUIGraphics](#) object. Handler takes one argument, a [DrawState](#) object.

void onMove ()

An event-handler callback function, called when the window has been moved

void onMoving ()

An event-handler callback function, called when the window is being moved
Called while a window is being moved, each time the position changes. A handler can monitor the move operation.

void onResize ()

An event-handler callback function, called after the window has been resized

void onResizing ()

An event-handler callback function, called while a window is being resized
Called while a window is being resized, each time the height or width changes. A handler can monitor the resize operation.

void onShortcutKey ()

In Windows only, an event-handler callback function, called a shortcut-key sequence is typed that matches the [shortcutKey](#) value for this window.

void onShow ()

An event-handler callback function, called just before the window is displayed

Called when a request is made to open the window using the `show()` method, before the window is made visible, but after automatic layout is complete. A handler can modify the results of the automatic layout.

String prompt (`prompt:String`, `default:String`, `title:String`)

Displays a modal dialog that returns the user's text input.

Returns the value of the text edit field if the user clicked OK, null if the user clicked Cancel.

Parameter	Type	Description
<code>prompt</code>	String	The string for the displayed message.
<code>default</code>	String	The initial value to be displayed in the text edit field.
<code>title</code>	String	A string to appear as the title of the dialog. In Windows, this appears in the window's frame; in Mac OS it appears above the message. The default title string is "Script Prompt".

void remove (`what:Any`)

Removes the specified child control from this window's `children` array.

No error results if the child does not exist.

Parameter	Type	Description
<code>what</code>	Any	The child control to remove, specified by 0-based index, text property value, or as a control object.

bool removeEventListener (`eventName:String`, `handler:Function` [, `capturePhase:bool=false`])

Unregisters an event handler for a particular type of event occurring in this window.

All arguments must be identical to those that were used to register the event handler.

Parameter	Type	Description
<code>eventName</code>	String	The name of the event.
<code>handler</code>	Function	The function that handles the event.
<code>capturePhase</code>	bool	Whether to call the handler only in the capturing phase of the event propagation. (default: false)

void show ()

Makes this window visible.

If an `onShow` callback is defined for a window, calls that function before showing the window. When a window or container is hidden, its children are also hidden, but when it is shown again, the children retain their own visibility states.

- For a modal dialog, opens the dialog and does not return until the dialog is dismissed. If it is dismissed via the `close()` method, this method returns any result value passed to that method. Otherwise, returns 0.

Element of

[Button.window](#)

[Checkbox.window](#)

[DropDownList.window](#)

[EditText.window](#)

[FlashPlayer.window](#)

[Group.window](#)

[IconButton.window](#)

[Image.window](#)

[ListBox.window](#)

[Panel.window](#)

[Progressbar.window](#)

[RadioButton.window](#)

[Scrollbar.window](#)

[Slider.window](#)

[StaticText.window](#)

[TreeView.window](#)

[Window.window](#)

Used in

[Window.center](#) ([window:Window](#))

Return

Window [Window.find](#) ([type:String](#), [title:String](#))

[Contents](#) :: [Index](#)

Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Bounds	LayoutManager	ScriptUIImage
Button	ListBox	ScriptUIPath
Checkbox	ListItems	ScriptUIPen
Dimension	Panel	Scrollbar
DrawState	Point	Slider
DropDownList	Progressbar	StaticText
EditText	RadioButton	TreeView
FlashPlayer	ScriptUI	UIEvent
Group	ScriptUIBrush	Window
IconButton	ScriptUIFont	
Image	ScriptUIGraphics	