



UIML Reference Manual

December 1997

Harmonia

Creating Harmony Between People and Computers

If you read this document, then you agree to keep the document confidential for three years, with the exclusion that prior knowledge and work already in progress at your company are not within the scope of this agreement should they overlap. You may disclose the document to a small number of your colleagues to evaluate your company's interest in Harmonia's products.

UIML Reference Manual

Introduction

This document is a reference manual for the User Interface Markup Language (UIML). It is divided into these sections:

1. This introduction
2. User interface definition tags
3. Style definition tags
 - a. Renderings - lists the individual user interface components available through UIML.
 - b. Components - lists the user interface components and the attributes used with each.
 - c. Attributes - lists attributes alphabetically and allowable values for each.

This document is intended as a reference for the language. It assumes familiarity with the structure and syntax of UIML. For a comprehensive description of UIML, see the separate tutorial.

Overview of UIML

UIML has three distinct components, with their own purposes:

1. The *user interface definition* specifies the content and structure of the user interface in a device-independent manner. It describes instances of user interface elements (or components) as well as classes of components.
2. The *style definition* specifies device-dependent characteristics for *classes* of user interface components. Note the style definition doesn't specify characteristics of instances of user interface components; use the content database for this. Rather, the style definition says, for example, that all of a certain class of buttons are to be rendered in a certain size with certain colors.
3. The *content database* specifies sets of content for specific user interface elements. For example, the content database might specify the label to use for a specify button in several languages.

This structure means that you can create your user interface in the user interface definition; you'll have one of these for your application (unless you have several interfaces for a single application). Then, you'll use the style definition to specify how to render user interface components on each of the platforms your application will support (personal computer, network computer, Internet appliance, etc.). Then, by matching the user interface definition to a style definition, you can create the user interface for that device.

It's generally good form to keep as much specific content as possible in the content database. Why? That's so sets of content can be easily swapped; for example, to translate your user interface into another language.

Note that attributes may be defined in the style definition or the content database. The difference is that in the style definition, attributes apply to *classes* of user interface components. In the content database, attributes apply to *individual* user interface components.

GROUPs and ELEMs -- What's The Difference?

All user interface objects as described in UIML as being either a GROUP or an ELEM ("element"). Groups are containers; that is, user interface objects that contain other user interface objects. Examples of groups are frames, panels, and scrollpanes.

Elms are individual user interface objects that do not contain other objects. Examples are buttons, lists and textfields.

Groups must be rendered as containers, but they may be empty. Elms can be any component, whether a container or not. But elms cannot contain anything.

The special type, "None", is used to disable rendering for a class of components. This means you can describe a complete user interface in your user interface definition, and build selected subsets by only modifying the style definition. Simply render the user interface objects you *don't* want to include as "None" (see the description of the RENDER attribute later in this document). "None" can behave as either a group or elem.

Attribute Scoping

By default, attribute values apply only to the component on which they're defined. However, attributes may be "scoped"; that is, marked so the attribute values are carried to all user interface components contained within the component that defined the attribute.

Attribute scoping is controlled by a prefix in front of the attribute name. In the prerelease, scoping prefixes are implemented as follows:

Prefix	Effect on scoping
+	Scoped: all components contained within the current component (<i>children</i>) receive this value.
-	Local override: this attribute value applies only to the current component and overrides any attribute value scoped by a parent container. (NOTE: <i>whether children receive attribute values scoped from parents depends on the attribute. This problem will be fixed in a future version. Meanwhile, attribute assignments can be precisely controlled by specifying attribute values at all levels.</i>)
None	Same as "-". ¹

For example, given a user interface consisting of a button within a frame, as follows:

¹ In the prerelease version, RENDERING-PREFIX must be preceded by a "-" for its value to be non-scoped.

```
<APP CLASS="Application" NAME="New Generator Test">
  <GROUP CLASS="Frame" NAME="aFrame">
    <ELEM CLASS="Button" NAME="aButton"/>
  </GROUP>
</APP>
```

The background colors of both the frame and button can be set to gray in the style definition as follows:

```
APP.Application {
  +TOOLKIT:jfc;
  +RENDERING-PREFIX:java.awt;
}

GROUP.Frame{
  RENDERING:Frame;
  BACKGROUND:gray;
}

ELEM.Button{
  RENDERING:Button;
  BACKGROUND:gray;
}
```

However, the background color can be defined for all components in the application by placing a "+" in front of the attribute name, eliminating the need to explicitly set the frame's and button's background colors:

```
APP.Application {
  +TOOLKIT:jfc;
  +RENDERING-PREFIX:java.awt;
  +BACKGROUND:gray;
}

GROUP.Frame{
  RENDERING:Frame;
}

ELEM.Button{
  RENDERING:Button;
}
```

Be careful only to put attributes that are common to all elements of a class in the style definition. Attributes that go with specific user interface elements (instances) should go into the content database. For example, consider the following style definition:

```
GROUP.AppWindow {
  RENDERING: Frame;
  LAYOUT: FlowLayout;
  SIZE: 525,530;
  BACKGROUND: "lightGray";
  FOREGROUND: "blue";
}
```

By defining the background and foreground colors in at the group level, all interface components in the window will also have the same colors. But what about size? By defining size at the group level, everything in the window will be rendered at the same size ... even the buttons!

Note the TOOLKIT and RENDERING-PREFIX attributes are also "scoped." That's so the renderer knows that all user interface components contained in the frame are also to be rendered from the Java™² Foundation Class's ("jfc") java.awt toolkit.

TOOLKIT and RENDERING-PREFIX

The TOOLKIT and RENDERING-PREFIX must appear as APP attributes.

Actually, it is very logical to put both TOOLKIT and RENDERING-PREFIX into the <APP> tag's style, since that will allow them to be scoped to all frames. This means you don't have to repeat them for each frame in the application.

Normally, TOOLKIT and RENDERING-PREFIX are scoped so other user interface elements receive their values. But they can be non-scoped if you don't plan to use them elsewhere.

Thus, in your user interface definition, specify:

```
<APP NAME="ApplicationName" CLASS="App">
```

and in the style definition:

```
APP.App {  
    +TOOLKIT: jfc;  
    +RENDERING-PREFIX:java.awt;  
}
```

To use other user interface toolkits, specify the toolkit and rendering-prefix accordingly. For example, to use Swing, you'd specify *com.sun.java.swing* as the RENDERING-PREFIX. To use a mix of renderings, use -RENDERING-PREFIX:java.awt and specify the full class path for all renderings (e.g., RENDERING:com.sun.java.swing.Jbutton or RENDERING:java.awt.Button). (The "-" prefix indicates this attribute value applies only to the application container; see discussion of attribute scoping below.)

Note that the prerelease version supports only the java.awt toolkit. However, this version includes a preliminary implementation of some Swing components.

User Interface Definition Semantics

When rendering with the java.awt (or swing) toolkit, user interface components are added to their containers in the order in which their tags appear in the user interface definition. Thus, if the user interface definition assigns two components to the same region in a container, the component defined later overwrites the one defined earlier.

For example, consider java.awt.BorderLayout. It allows just one component to be assigned to the North, East, South, West, and Center regions. Thus, in the following user interface definition, "Button2" is

² Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

placed on top of "Button1", so "Button1" is useless (and might as well be removed from the user interface definition):

```
<GROUP CLASS="Panel"> <!-- LAYOUT:BorderLayout -->
  <ELEM CLASS="Button" NAME="Button1"> <!--ALIGNMENT=Center->
  <ELEM CLASS="Button" NAME="Button2"> <!--ALIGNMENT=Center->
</GROUP>
```

It is possible to construct a user interface definition with java.awt in which several components occupy the same region in a container. This is done by using LAYOUT:CardLayout with a Panel. CardLayout is planned for implementation in a future version.

Attributes Semantics

Once the user interface is built by adding interface components to their containers, the attributes of all components are set according to the style definition and content database. At this point, the interface is rendered, and the user can interact with it.

These interactions cause the execution of ACTION tags. ACTION tags, which are defined in the user interface definition, simply change the values of attributes of other interface components, and then cause the interface to be repainted (in java.awt). Repainting retains the component nesting order defined in the user interface definition.

When an ACTION tag causes the interface to be repainted, the order in which components appear is also governed by the VISIBLE attribute. User interface components can be made visible or invisible by setting the VISIBLE attribute to *true* or *false*, respectively.

After the initial rendering of the interface, VISIBLE attributes are set in the order in which ACTION tags are executed. Within each ACTION, VISIBLE attributes are set in the order in which they appear in the ACTION tag's VALUE= field.

When the interface is repainted, if two components occupy the same region, the one on top is the one whose VISIBLE attribute was last set to *true*.

Allowable Characters and Case Sensitivity

In the prerelease version, all attribute names must be in upper case. Attribute values are case-sensitive and must match the case shown in this document.

All names (group names, element names, class names, etc.) must follow the syntax:

```
[A-Za-z_][A-Za-z0-9_]*
```

That is, the first character must be a letter or an underscore, and subsequent characters may be letters, numbers, or an underscore.

In style definitions, attribute values must be enclosed in double quotes if they contain special characters with syntactic significance, such as commas and colons.

Command Line Syntax

From a command prompt, use **java Render** with no options to see the command line format and options.

Note that the command line syntax permits multiple style definitions. Style definitions are processed in the order they appear in the command line. If a component's attribute is defined in more than one style definition, the last definition will override any earlier definitions.

Color Table

Several attributes, including BACKGROUND and FOREGROUND, allow specifying colors in several ways. The following table is a guide to some common colors.

Color	Red	Green	Blue	Hue	Saturation	Brightness
black	0	0	0	0	0	0
blue	0	0	255	.666667	1	1
cyan	0	255	255	.5	1	1
darkGray	64	64	64	0	0	.25098
gray	128	128	128	0	0	.501961
green	0	255	0	.333333	1	1
lightGray	192	192	192	0	0	.752941
magenta	255	0	255	.833333	1	1
orange	255	200	0	.130719	1	1
pink	255	175	175	0	.313726	1
red	255	0	0	0	1	1
white	255	255	255	0	0	1
yellow	255	255	0	.166667	1	1

UIML Language Reference

User Interface Definition Tags

TagName: ACTION

Syntax: <ACTION
 TRIGGER=trigger
 VALUE="value-list"
/>

Allowed SubTags:

Description:

Used to allow one user interface element to cause a change in state in another user interface element.

ACTION tags define message passing between user interface elements, without calling the back-end application. Use the CALL tag to define events that result calls to the back-end application.

ACTIONS are executed by changing attribute values of user interface elements. In the prerelease version, ACTION supports the following attributes for java.awt components: ALIGNMENT-HORIZONTAL, BACKGROUND, COLUMNS, CONTENT, EDITABLE, ENABLED, FOREGROUND, LAYOUT, LEFT, LOCATION, MULTIPLEMODES, ROWS, SIZE, STATE, TOP, VISIBLE.

Attributes:

Attribute	AllowedValues	Meaning
TRIGGER	Activated, Closed, Closing, Deactivated, Deiconified, Iconified, Opened, Selected, Deselected, Selected.integer	<p>The name of the trigger that causes this action to take place. Triggers are:</p> <ul style="list-style-type: none"> - Activated, Deactivated: An element is activated when it receives focus (is ready to act on events). It's deactivated when it loses focus. Deactivated elements usually change their appearance; for example, a menu item might be greyed-out when it's deactivated. - Opened, Closed: An element is opened when it appears as part of the interface; when closed, it is removed from the user interface and its resources are freed. (Use Closed only if other open frames are on the screen; otherwise the application will exit before executing the ACTION tag.) - Closing - The frame is about to be closed, but hasn't closed yet. This trigger is useful for actions that must happen (such as saving contents) before a window closes. - Iconified, Deiconified: a window is iconified when it is removed from the screen and represented solely by an icon on the desktop. In Windows, this action is called "minimize". A window is deiconified when it is restored to the screen as an open window. - Selected: "Selected" means clicking on a button, selecting a menu item, double-clicking on a list entry, or completing text editing in a text field (for example, by pressing the Enter key). - Selected.integer: In Choice and List components, the entry

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

Attribute	AllowedValues	Meaning
		specified by <integer> was selected. (Note: in the prerelease version, the ability of ACTION to work with a List whose MULTIPLEMODE attribute is "true" has not yet been implemented.)
VALUE	<p>A set of value clauses, separated by semicolons. Each value clause is of the form <name>.<attribute name>=<value>, where:</p> <ul style="list-style-type: none">- <name> is the name of a user interface element defined in an APP, GROUP, or ELEM tag;- <attribute name> is the name of an attribute associated with the user interface element; and- <value> is a new value for the attribute.	The action to be carried out on other user interface elements. Value clauses are executed in left to right order. Note that all NAME= fields in a single .ui file must be unique.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: APP

Syntax: <APP>
.. UIML statements ..
</APP>

Allowed SubTags: ELEM, GROUP, PROPERTIES

Description:

Describes the overall structure of the user interface.

Attributes:

Attribute	AllowedValues	Meaning
CLASS	A class name described in the style file.	Which portrayal, from the style definition, to use for the user interface. This attribute must be specified.
NAME	A string	A name for the application

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: AUTHOR

Syntax: <AUTHOR>
.. text ..
</AUTHOR>

Allowed SubTags:

Description:

Use this tag to record who authored the UIML document.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: DATE

Syntax: <DATE>
.. text ..
</DATE>

Allowed SubTags:

Description:

Use this tag to record the interface definition's date. The date is just text; enter it in any convenient format between the opening and closing tags.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: DEFINE

Syntax: <DEFINE>
.. UIML statements ..
</DEFINE>

Allowed SubTags: ELEM, GROUP, PROPERTIES

Description:

Defines the details of a specific user interface element, group, or application.

The element in the NAME parameter must have been defined earlier in the user interface definition. Otherwise the definition is ignored.

The DEFINE tag may contain GROUP and ELEM tags that define additional user interface elements that are a part of the group specified in the NAME parameter. This means you may define high-level constructs up front in the user interface definition, then specify the details later in a DEFINE section. For example, you might define a menu up front, then define the menu's options later in a DEFINE section.

Properties common to user interface elements described within a DEFINE tag may be specified once at the beginning of the tag. For example:

```
<DEFINE NAME= "MainMenu">  
  <PROPERTIES>  
    ... common properties here ...  
  </PROPERTIES>  
  <GROUP NAME= "FileMenu">  
    <PROPERTIES>  
      ... properties specific to "FileMenu" here ...  
    </PROPERTIES>  
  </GROUP>  
</DEFINE>
```

Attributes:

Attribute	AllowedValues	Meaning
NAME	A string	Identifies the element being defined. This must match the name of a user interface element, group, or application defined in an ELEM, GROUP, or APP tag that appears earlier in the user interface definition.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: DEFINITIONS

Syntax: <DEFINITIONS
 FILENAME="file-name"
/>

Allowed SubTags:

Description:

(Not implemented in the prerelease version.)

Specifies the name of a file containing user interface element definitions.

UIML documents are easier to read if broken into two sections: the first enumerating major user interface elements, and the second defining the details of the user interface elements. The DEFINITIONS tag allows putting the definitions in a separate file.

Attributes:

Attribute	AllowedValues	Meaning
FILENAME	A string	Name of a file containing definitions.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: ELEM

Syntax: <ELEM
 NAME="element-name"
 CLASS="class-name"
/>

Allowed SubTags:

Description:

Defines a single user interface element, such as a button, text field, or combo box in a graphical user interface.

Use ELEM to define individual user interface elements. Use GROUP to define compound elements, i.e., elements that contain other elements.

Syntactically, the NAME and CLASS parameters are optional. CLASS may be specified in a DEFINE tag.

Attributes:

Attribute	AllowedValues	Meaning
CLASS	A string	Identifies which portrayal, from the style definition, to use for the user interface element.
NAME	A string	Identifies the user interface element.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: GROUP

Syntax: <GROUP>
.. UIML statements ..
</GROUP>

Allowed SubTags: ELEM, GROUP, PROPERTIES

Description:

Defines a compound user interface element; i.e., elements that contain other elements. Examples of compound interface elements are frames, menus, and panels.

Groups are most useful in two cases:

1. to define a set of identical user interface elements with related purposes, such as groups of buttons in a graphical user interface; and
2. to define a set of user interface elements of different types that receive data from a common source or from a single call to the back-end application.

Groups may be nested; for example, to define submenus within menus.

Syntactically, the NAME and CLASS parameters are optional. CLASS may be specified in a DEFINE tag.

Attributes:

Attribute	AllowedValues	Meaning
CLASS	A string	Identifies which portrayal, from the style definition, to use for the group.
NAME	A string	Identifies the group.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: HEAD

Syntax: <HEAD>
.. UIML statements ..
</HEAD>

Allowed SubTags: TITLE, AUTHOR, DATE, VERSION

Description:

Contains information about the user interface that's not displayed in the interface itself.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: PROPERTIES

Syntax: <PROPERTIES>
.. UIML statements ..
</PROPERTIES>

Allowed SubTags: ACTION, CALL, SOURCE

Description:

Enumerates the characteristics of a user interface element, such as events, calls to the back end application, etc. PROPERTIES is normally used within a DEFINE tag.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: TITLE

Syntax: <TITLE>
.. text ..
</TITLE>

Allowed SubTags:

Description:

Records a title for the user interface definition or application.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: UIML

Syntax: <UIML>
.. UIML statements ..
</UIML>

Allowed SubTags: APP, DEFINE, DEFINITIONS, HEAD

Description:

Defines the boundaries of the user interface definition. All UIML statements, including comments, must appear between <UIML> and </UIML>. <UIML> and </UIML> are normally the first and last lines, respectively, of a user interface definition.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

User Interface Definition Tags

TagName: VERSION

Syntax: <VERSION>
.. text ..
</VERSION>

Allowed SubTags:

Description:

Records a version number for the user interface definition.

Attributes:

Attribute	AllowedValues	Meaning
(none)		No attributes defined for this tag.

Note: Attribute names and values are case sensitive.

UIML Language Reference

UIML Renderings

RENDERING

AllowedValues	Meaning
Button, Choice, Label, List, Panel, ScrollPane, TextArea, TextField	Inserts a java.awt component into a parent component. The parent component must be rendered as a Frame, Panel, or ScrollPane.
ButtonGroup	JRadioButtons nested in a ButtonGroup can be checked with mutual exclusion. The parent component must be rendered as a Frame, Panel, ScrollPane, JFrame, or JPanel.
Checkbox	Inserts a Checkbox into a parent component. The parent component must be rendered as a CheckboxGroup, Frame, Panel, or ScrollPane.
CheckboxGroup	Transforms Checkboxes into radio buttons. Place Checkboxes inside a CheckboxGroup to convert them to radio buttons. The parent component must be rendered as a Frame, Panel, or ScrollPane.
Frame	Inserts a Frame into a parent component. The parent must be an APP tag.
JButton, JCheckBox, JPanel, JRadioButton, JToggleButton	Inserts a Java Swing component into a parent component. The parent component must be rendered as a frame, panel, scrollpane, JFrame, or JPanel.
JFrame	Inserts a JFrame into a parent component. The parent must be an APP tag.
Menu	Inserts a Menu into the parent component's menubar. The Menu must be nested one or more levels within a Frame.
MenuItem	Inserts a MenuItem into a menu. The parent component must be rendered as a Menu.
None	Ignores the RENDERING tag. Useful for removing user interface elements for certain devices without having to modify the user interface definition itself (the .ui file). Note: any components contained within the component rendered as None will still be rendered.
TopPanel	Create a Panel that is not contained in any other container. Used as top level container in Active-X version of renderer.

UIML Language Reference

UIML Interface Components

RENDERING:Button

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Button

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Button

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Button

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
VISIBLE	true, false	<p>Specifies whether to show the user interface component (true shows; false hides). The default is true.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Button

	AllowedValues	Meaning
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Button

	AllowedValues	Meaning
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Checkbox

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Checkbox

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Checkbox

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Checkbox

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
STATE	true, false	If true, the checkbox is checked when initially rendered.
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
VISIBLE	true, false	Specifies whether to show the user interface component (true shows; false hides). The default is true.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Checkbox

	AllowedValues	Meaning
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Checkbox

	AllowedValues	Meaning
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:CheckboxGroup

AllowedValues	Meaning
No attributes enabled for this component.	

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Choice

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Choice

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Choice

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Choice

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
VISIBLE	true, false	<p>Specifies whether to show the user interface component (true shows; false hides). The default is true.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Choice

	AllowedValues	Meaning
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Choice

	AllowedValues	Meaning
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Frame

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Frame

	AllowedValues	Meaning
COLUMNS	A positive integer	In GridLayout: exact number of columns to use. In this usage, COLUMNS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea and TextField: width of the component, in characters. In this usage, COLUMNS is specified as an attribute of the component.
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
EXISTS	true, false	If false: - App tag: terminate the Renderer. - RENDERING:Frame: destroys frame. If no visible frames remain, terminate Renderer If true: no effect. (Default is true.)
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Frame

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
HGAP	A non-negative integer	The amount of space between each component in the container and the component immediately to its left or right. (Note: This is not the amount of space around an interface component.) HGAP is specified as an attribute of the container.
LAYOUT	AbsoluteLayout, BorderLayout, FlowLayout, GridLayout, GridBagLayout	See AWT documentation. FlowLayout is default for Panel. BorderLayout is default for Frame. AbsoluteLayout is the equivalent of a "null" LayoutManager in AWT. Note with AbsoluteLayout, component positions and size must be specified. Use LOCATION or LEFT and TOP to specify position; use SIZE to specify component size.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Frame

	AllowedValues	Meaning
LEFT	A non-negative integer.	<p>Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
ROWS	A positive integer	In GridLayout: exact number of rows to use. In this usage, ROWS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea: height of the component, in rows. In this usage, ROWS is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Frame

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's size, in pixels. This attribute is implemented: - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. Otherwise SIZE is a preference that the layout manager may or may not elect to use. Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.
TOP	A non-negative integer	Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.
VGAP	A non-negative integer	The amount of space between each component in the container and the component immediately above or below. (Note: This is not the amount of space around an interface component.) VGAP is specified as an attribute of the container.
VISIBLE	true, false	Specifies whether to show the user interface component (true shows; false hides). The default is true.
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Frame

	AllowedValues	Meaning
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs.</p> <p>XWEIGHT is specified as an attribute of the component.</p>
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JButton

	AllowedValues	Meaning
ALIGNMENT-HORIZONTAL	Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.
ALIGNMENT-VERTICAL	Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center
AUTOSCROLLS	true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.
BORDER	LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
	RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
	LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
	CompoundBorder	Combination of the LineBorders.
	TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.
BORDER-PAINTED	true, false	Sets whether the border should be painted. Default is true.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FOCUS-PAINTED	true, false	If true, then focus is painted. (After you click on the button, a dotted line around the button shows the button has the focus. This line disappears when you click somewhere else, showing the button has lost the focus.) Default is false.
IMAGE	Filename or URL	Sets the icon to display on a button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JButton

	AllowedValues	Meaning
IMAGE-DISABLED	Filename or URL	Sets the icon to display when the button is disabled. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-DISABLED-SELECTED	Filename or URL	Sets the icon to display when the button is disabled and selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-PRESSED	Filename or URL	Sets the icon to display when the button is pressed. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER	Filename or URL	Sets the icon to display when the mouse is over the button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER-SELECTED	Filename or URL	Sets the icon to display when ROLLOVER effects are enabled and the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-SELECTED	Filename or URL	Sets the icon to display when the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JButton

	AllowedValues	Meaning
LINEBORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Sets the color of the LineBorder's line to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set the color of the border's line to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A positive integer	Set the color of the border's line to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set the color of the border's line to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
LINEBORDER-THICKNESS	An integer	Sets the thickness of the LineBorder's line.
OPAQUE	true, false	If true, the component's background will be filled with the background color.
ROLLOVER	true, false	Sets whether rollover effects should be enabled. Rollover effects allow icons to change when the mouse is on top of a button. You must also define an IMAGE-ROLLOVER.
SELECTED	true, false	Sets the state of a button.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JButton

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's size, in pixels. This attribute is implemented: - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. Otherwise SIZE is a preference that the layout manager may or may not elect to use. Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.
SIZE-MAXIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.
SIZE-MINIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.
TEXT-HORIZONTAL	Center, Left, Right	Sets the horizontal position of text relative to an icon. Default is Left.
TEXT-VERTICAL	Center, Top, Bottom	Sets the vertical position of text relative to an icon. Default is Center.
TITLED BORDER-BORDER	LineBorder, RaisedBevelBorder, LoweredBevelBorder, CompoundBorder	Specifies what line style to use for a TitledBorder.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JButton

	AllowedValues	Meaning
TITLED BORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set titled-border color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set titled-border color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	An integer	Set titled-border color to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
TITLED BORDER-FONT-NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
TITLED BORDER-FONT-SIZE	A positive integer	Font size, in points.
TITLED BORDER-FONT-STYLE	plain, bold, italic, bold italic	Font style (bold, italic, etc.)
TITLED BORDER-JUSTIFICATION	Left, Center, Right	Sets whether the title should appear on the left, center, or right. Default is Left.
TITLED BORDER-POSITION	Above-Top, Top, Below-Top, Above-Bottom, Bottom, Below-Bottom	Sets whether the title should appear above, on, or below the top or bottom line of the border. Default is Top.
TITLED BORDER-TITLE	A string	Sets the title to place on the title-border.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JButton

AllowedValues	Meaning
---------------	---------

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JCheckBox

	AllowedValues	Meaning
ALIGNMENT-HORIZONTAL	Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.
ALIGNMENT-VERTICAL	Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center
AUTOSCROLLS	true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.
BORDER	LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
	RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
	LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
	CompoundBorder	Combination of the LineBorders.
	TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.
BORDER-PAINTED	true, false	Sets whether the border should be painted. Default is true.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FOCUS-PAINTED	true, false	If true, then focus is painted. (After you click on the button, a dotted line around the button shows the button has the focus. This line disappears when you click somewhere else, showing the button has lost the focus.) Default is false.
IMAGE	Filename or URL	Sets the icon to display on a button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JCheckBox

	AllowedValues	Meaning
IMAGE-DISABLED	Filename or URL	Sets the icon to display when the button is disabled. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-DISABLED-SELECTED	Filename or URL	Sets the icon to display when the button is disabled and selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-PRESSED	Filename or URL	Sets the icon to display when the button is pressed. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER	Filename or URL	Sets the icon to display when the mouse is over the button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER-SELECTED	Filename or URL	Sets the icon to display when ROLLOVER effects are enabled and the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-SELECTED	Filename or URL	Sets the icon to display when the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JCheckBox

	AllowedValues	Meaning
LINEBORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Sets the color of the LineBorder's line to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set the color of the border's line to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A positive integer	Set the color of the border's line to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set the color of the border's line to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none">· Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black).· A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color.· A brightness of 0 is black; a brightness of 1 is maximum color saturation.
LINEBORDER-THICKNESS	An integer	Sets the thickness of the LineBorder's line.
OPAQUE	true, false	If true, the component's background will be filled with the background color.
ROLLOVER	true, false	Sets whether rollover effects should be enabled. Rollover effects allow icons to change when the mouse is on top of a button. You must also define an IMAGE-ROLLOVER.
SELECTED	true, false	Sets the state of a button.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JCheckBox

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's size, in pixels. This attribute is implemented: - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. Otherwise SIZE is a preference that the layout manager may or may not elect to use. Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.
SIZE-MAXIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.
SIZE-MINIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.
TEXT-HORIZONTAL	Center, Left, Right	Sets the horizontal position of text relative to an icon. Default is Left.
TEXT-VERTICAL	Center, Top, Bottom	Sets the vertical position of text relative to an icon. Default is Center.
TITLED BORDER-BORDER	LineBorder, RaisedBevelBorder, LoweredBevelBorder, CompoundBorder	Specifies what line style to use for a TitledBorder.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JCheckBox

	AllowedValues	Meaning
TITLED BORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set titled-border color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set titled-border color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	An integer	Set titled-border color to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
TITLED BORDER-FONT-NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
TITLED BORDER-FONT-SIZE	A positive integer	Font size, in points.
TITLED BORDER-FONT-STYLE	plain, bold, italic, bold italic	Font style (bold, italic, etc.)
TITLED BORDER-JUSTIFICATION	Left, Center, Right	Sets whether the title should appear on the left, center, or right. Default is Left.
TITLED BORDER-POSITION	Above-Top, Top, Below-Top, Above-Bottom, Bottom, Below-Bottom	Sets whether the title should appear above, on, or below the top or bottom line of the border. Default is Top.
TITLED BORDER-TITLE	A string	Sets the title to place on the title-border.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JCheckBox

AllowedValues	Meaning
---------------	---------

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JLabel

	AllowedValues	Meaning
ALIGNMENT-HORIZONTAL	Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.
ALIGNMENT-VERTICAL	Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center
AUTOSCROLLS	true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.
BORDER	LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
	RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
	LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
	CompoundBorder	Combination of the LineBorders.
	TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none"> - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
SIZE-MAXIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JLabel

	AllowedValues	Meaning
SIZE-MINIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JPanel

	AllowedValues	Meaning
AUTOSCROLLS	true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.
BORDER	LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
	RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
	LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
	CompoundBorder	Combination of the LineBorders.
	TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.
OPAQUE	true, false	If true, the component's background will be filled with the background color.
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none"> - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
SIZE-MAXIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.
SIZE-MINIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JPanel

	AllowedValues	Meaning
TITLED BORDER-BORDER	LineBorder, RaisedBevelBorder, LoweredBevelBorder, CompoundBorder	Specifies what line style to use for a TitledBorder.
TITLED BORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set titled-border color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set titled-border color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	An integer	Set titled-border color to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
TITLED BORDER-FONT-NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
TITLED BORDER-FONT-SIZE	A positive integer	Font size, in points.
TITLED BORDER-FONT-STYLE	plain, bold, italic, bold italic	Font style (bold, italic, etc.)
TITLED BORDER-JUSTIFICATION	Left, Center, Right	Sets whether the title should appear on the left, center, or right. Default is Left.
TITLED BORDER-POSITION	Above-Top, Top, Below-Top, Above-Bottom, Bottom, Below-Bottom	Sets whether the title should appear above, on, or below the top or bottom line of the border. Default is Top.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JPanel

	AllowedValues	Meaning
TITLEDDBORDER-TITLE	A string	Sets the title to place on the title-border.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JRadioButton

	AllowedValues	Meaning
ALIGNMENT-HORIZONTAL	Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.
ALIGNMENT-VERTICAL	Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center
AUTOSCROLLS	true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.
BORDER	LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
	RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
	LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
	CompoundBorder	Combination of the LineBorders.
	TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.
BORDER-PAINTED	true, false	Sets whether the border should be painted. Default is true.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FOCUS-PAINTED	true, false	If true, then focus is painted. (After you click on the button, a dotted line around the button shows the button has the focus. This line disappears when you click somewhere else, showing the button has lost the focus.) Default is false.
IMAGE	Filename or URL	Sets the icon to display on a button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JRadioButton

	AllowedValues	Meaning
IMAGE-DISABLED	Filename or URL	Sets the icon to display when the button is disabled. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-DISABLED-SELECTED	Filename or URL	Sets the icon to display when the button is disabled and selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-PRESSED	Filename or URL	Sets the icon to display when the button is pressed. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER	Filename or URL	Sets the icon to display when the mouse is over the button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER-SELECTED	Filename or URL	Sets the icon to display when ROLLOVER effects are enabled and the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-SELECTED	Filename or URL	Sets the icon to display when the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JRadioButton

	AllowedValues	Meaning
LINEBORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Sets the color of the LineBorder's line to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set the color of the border's line to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A positive integer	Set the color of the border's line to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set the color of the border's line to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
LINEBORDER-THICKNESS	An integer	Sets the thickness of the LineBorder's line.
OPAQUE	true, false	If true, the component's background will be filled with the background color.
ROLLOVER	true, false	Sets whether rollover effects should be enabled. Rollover effects allow icons to change when the mouse is on top of a button. You must also define an IMAGE-ROLLOVER.
SELECTED	true, false	Sets the state of a button.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JRadioButton

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's size, in pixels. This attribute is implemented: - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. Otherwise SIZE is a preference that the layout manager may or may not elect to use. Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.
SIZE-MAXIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.
SIZE-MINIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.
TEXT-HORIZONTAL	Center, Left, Right	Sets the horizontal position of text relative to an icon. Default is Left.
TEXT-VERTICAL	Center, Top, Bottom	Sets the vertical position of text relative to an icon. Default is Center.
TITLED BORDER-BORDER	LineBorder, RaisedBevelBorder, LoweredBevelBorder, CompoundBorder	Specifies what line style to use for a TitledBorder.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JRadioButton

	AllowedValues	Meaning
TITLED BORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set titled-border color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set titled-border color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	An integer	Set titled-border color to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
TITLED BORDER-FONT-NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
TITLED BORDER-FONT-SIZE	A positive integer	Font size, in points.
TITLED BORDER-FONT-STYLE	plain, bold, italic, bold italic	Font style (bold, italic, etc.)
TITLED BORDER-JUSTIFICATION	Left, Center, Right	Sets whether the title should appear on the left, center, or right. Default is Left.
TITLED BORDER-POSITION	Above-Top, Top, Below-Top, Above-Bottom, Bottom, Below-Bottom	Sets whether the title should appear above, on, or below the top or bottom line of the border. Default is Top.
TITLED BORDER-TITLE	A string	Sets the title to place on the title-border.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JRadioButton

AllowedValues	Meaning
---------------	---------

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JToggleButton

	AllowedValues	Meaning
ALIGNMENT-HORIZONTAL	Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.
ALIGNMENT-VERTICAL	Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center
AUTOSCROLLS	true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.
BORDER	LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
	RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
	LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
	CompoundBorder	Combination of the LineBorders.
	TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.
BORDER-PAINTED	true, false	Sets whether the border should be painted. Default is true.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FOCUS-PAINTED	true, false	If true, then focus is painted. (After you click on the button, a dotted line around the button shows the button has the focus. This line disappears when you click somewhere else, showing the button has lost the focus.) Default is false.
IMAGE	Filename or URL	Sets the icon to display on a button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JToggleButton

	AllowedValues	Meaning
IMAGE-DISABLED	Filename or URL	Sets the icon to display when the button is disabled. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-DISABLED-SELECTED	Filename or URL	Sets the icon to display when the button is disabled and selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-PRESSED	Filename or URL	Sets the icon to display when the button is pressed. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER	Filename or URL	Sets the icon to display when the mouse is over the button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER-SELECTED	Filename or URL	Sets the icon to display when ROLLOVER effects are enabled and the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-SELECTED	Filename or URL	Sets the icon to display when the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JToggleButton

	AllowedValues	Meaning
LINEBORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Sets the color of the LineBorder's line to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set the color of the border's line to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A positive integer	Set the color of the border's line to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set the color of the border's line to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
LINEBORDER-THICKNESS	An integer	Sets the thickness of the LineBorder's line.
OPAQUE	true, false	If true, the component's background will be filled with the background color.
ROLLOVER	true, false	Sets whether rollover effects should be enabled. Rollover effects allow icons to change when the mouse is on top of a button. You must also define an IMAGE-ROLLOVER.
SELECTED	true, false	Sets the state of a button.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JToggleButton

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's size, in pixels. This attribute is implemented: - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. Otherwise SIZE is a preference that the layout manager may or may not elect to use. Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.
SIZE-MAXIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.
SIZE-MINIMUM	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.
TEXT-HORIZONTAL	Center, Left, Right	Sets the horizontal position of text relative to an icon. Default is Left.
TEXT-VERTICAL	Center, Top, Bottom	Sets the vertical position of text relative to an icon. Default is Center.
TITLED BORDER-BORDER	LineBorder, RaisedBevelBorder, LoweredBevelBorder, CompoundBorder	Specifies what line style to use for a TitledBorder.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JToggleButton

	AllowedValues	Meaning
TITLED BORDER-COLOR	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set titled-border color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set titled-border color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	An integer	Set titled-border color to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
TITLED BORDER-FONT-NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
TITLED BORDER-FONT-SIZE	A positive integer	Font size, in points.
TITLED BORDER-FONT-STYLE	plain, bold, italic, bold italic	Font style (bold, italic, etc.)
TITLED BORDER-JUSTIFICATION	Left, Center, Right	Sets whether the title should appear on the left, center, or right. Default is Left.
TITLED BORDER-POSITION	Above-Top, Top, Below-Top, Above-Bottom, Bottom, Below-Bottom	Sets whether the title should appear above, on, or below the top or bottom line of the border. Default is Top.
TITLED BORDER-TITLE	A string	Sets the title to place on the title-border.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:JToggleButton

AllowedValues	Meaning
---------------	---------

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Label

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Label

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Label

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Label

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
VISIBLE	true, false	<p>Specifies whether to show the user interface component (true shows; false hides). The default is true.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Label

	AllowedValues	Meaning
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Label

	AllowedValues	Meaning
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:List

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:List

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:List

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:List

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
MULTIPLEMODE S	true, false	Indicates whether multiple items in a list box may be selected. Default is false (only one item may be selected).
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
VISIBLE	true, false	Specifies whether to show the user interface component (true shows; false hides). The default is true.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:List

	AllowedValues	Meaning
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs.</p> <p>XWEIGHT is specified as an attribute of the component.</p>
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:List

	AllowedValues	Meaning
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Menu

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:MenuItem

	AllowedValues	Meaning
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:None

AllowedValues	Meaning
No attributes enabled for this component.	

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Panel

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Panel

	AllowedValues	Meaning
COLUMNS	A positive integer	In GridLayout: exact number of columns to use. In this usage, COLUMNS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea and TextField: width of the component, in characters. In this usage, COLUMNS is specified as an attribute of the component.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Panel

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
HGAP	A non-negative integer	The amount of space between each component in the container and the component immediately to its left or right. (Note: This is not the amount of space around an interface component.) HGAP is specified as an attribute of the container.
LAYOUT	AbsoluteLayout, BorderLayout, FlowLayout, GridLayout, GridBagLayout	See AWT documentation. FlowLayout is default for Panel. BorderLayout is default for Frame. AbsoluteLayout is the equivalent of a "null" LayoutManager in AWT. Note with AbsoluteLayout, component positions and size must be specified. Use LOCATION or LEFT and TOP to specify position; use SIZE to specify component size.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Panel

	AllowedValues	Meaning
LEFT	A non-negative integer.	<p>Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
ROWS	A positive integer	In GridLayout: exact number of rows to use. In this usage, ROWS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea: height of the component, in rows. In this usage, ROWS is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Panel

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none"> - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
VGAP	A non-negative integer	The amount of space between each component in the container and the component immediately above or below. (Note: This is not the amount of space around an interface component.) VGAP is specified as an attribute of the container.
VISIBLE	true, false	Specifies whether to show the user interface component (true shows; false hides). The default is true.
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:Panel

	AllowedValues	Meaning
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.
YWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs. YWEIGHT is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:ScrollPane

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:ScrollPane

	AllowedValues	Meaning
COLUMNS	A positive integer	In GridLayout: exact number of columns to use. In this usage, COLUMNS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea and TextField: width of the component, in characters. In this usage, COLUMNS is specified as an attribute of the component.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:ScrollPane

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
HGAP	A non-negative integer	The amount of space between each component in the container and the component immediately to its left or right. (Note: This is not the amount of space around an interface component.) HGAP is specified as an attribute of the container.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:ScrollPane

	AllowedValues	Meaning
LEFT	A non-negative integer.	<p>Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
ROWS	A positive integer	In GridLayout: exact number of rows to use. In this usage, ROWS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea: height of the component, in rows. In this usage, ROWS is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:ScrollPane

	AllowedValues	Meaning
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's size, in pixels. This attribute is implemented: - for components whose container's LAYOUT is AbsoluteLayout, and - for Frames, regardless of layout. Otherwise SIZE is a preference that the layout manager may or may not elect to use. Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.
TOP	A non-negative integer	Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.
VGAP	A non-negative integer	The amount of space between each component in the container and the component immediately above or below. (Note: This is not the amount of space around an interface component.) VGAP is specified as an attribute of the container.
VISIBLE	true, false	Specifies whether to show the user interface component (true shows; false hides). The default is true.
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:ScrollPane

	AllowedValues	Meaning
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.
YWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs. YWEIGHT is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextArea

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextArea

	AllowedValues	Meaning
COLUMNS	A positive integer	In GridLayout: exact number of columns to use. In this usage, COLUMNS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea and TextField: width of the component, in characters. In this usage, COLUMNS is specified as an attribute of the component.
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
EDITABLE	true, false	Whether the text can be edited by the user. If true, the text can be edited (default).
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextArea

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextArea

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
SCROLLBARS	BOTH, HORIZONTAL_ONLY, NONE, VERTICAL_ONLY	<p>Specifies which scrollbars should be attached to the component. Default is BOTH. Using SCROLLBARS requires explicitly defining the component's CONTENT, COLUMNS, and ROWS attributes.</p>
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextArea

	AllowedValues	Meaning
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	If 0: Do not give the interface component extra space if window is resized. If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs. XWEIGHT is specified as an attribute of the component.
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextArea

	AllowedValues	Meaning
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextField

	AllowedValues	Meaning
ALIGNMENT	Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
	LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
	CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.
BACKGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextField

	AllowedValues	Meaning
COLUMNS	A positive integer	In GridLayout: exact number of columns to use. In this usage, COLUMNS is specified as an attribute of the container.
	A positive integer	In GridBagLayout: ignored.
	A positive integer	With TextArea and TextField: width of the component, in characters. In this usage, COLUMNS is specified as an attribute of the component.
CONTENT	A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.
EDITABLE	true, false	Whether the text can be edited by the user. If true, the text can be edited (default).
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FILL	NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.
FONT_NAME	Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).
FONT_SIZE	A positive integer	Font size, in points.
FONT_STYLE	PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextField

	AllowedValues	Meaning
FOREGROUND	black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
	One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
	0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
	A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
	<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.
HEIGHT	A positive integer	Interface component height, in rows.
	REMAINDER	Set component height to the number of rows remaining in the grid.
LEFT	A non-negative integer.	Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION. Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar. Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextField

	AllowedValues	Meaning
LOCATION	Two positive integers, separated by a comma (e.g., 125,73)	<p>Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
SIZE	Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>
TOP	A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>
WIDTH	A positive integer	Interface component width, in columns.
	REMAINDER	Set component width to the number of columns remaining in the grid.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:TextField

	AllowedValues	Meaning
XPAD	A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.
XPLACE	A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.
XWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs.</p> <p>XWEIGHT is specified as an attribute of the component.</p>
YPAD	A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.
YPLACE	A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
	RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.
YWEIGHT	A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:basic.BasicArrowButton

	AllowedValues	Meaning
ALIGNMENT-HORIZONTAL	Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.
ALIGNMENT-VERTICAL	Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center
BORDER-PAINTED	true, false	Sets whether the border should be painted. Default is true.
DIRECTION	North, South, East, West	Sets the direction of the arrow on the button.
ENABLED	true, false	If true, enables the component; otherwise, disables it. Default is true.
FOCUS-PAINTED	true, false	If true, then focus is painted. (After you click on the button, a dotted line around the button shows the button has the focus. This line disappears when you click somewhere else, showing the button has lost the focus.) Default is false.
IMAGE	Filename or URL	Sets the icon to display on a button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-DISABLED	Filename or URL	Sets the icon to display when the button is disabled. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-DISABLED-SELECTED	Filename or URL	Sets the icon to display when the button is disabled and selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-PRESSED	Filename or URL	Sets the icon to display when the button is pressed. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

Note: Attribute names and values are case sensitive.

UIML Interface Components

RENDERING:basic.BasicArrowButton

	AllowedValues	Meaning
IMAGE-ROLLOVER	Filename or URL	Sets the icon to display when the mouse is over the button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-ROLLOVER-SELECTED	Filename or URL	Sets the icon to display when ROLLOVER effects are enabled and the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
IMAGE-SELECTED	Filename or URL	Sets the icon to display when the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.
OPAQUE	true, false	If true, the component's background will be filled with the background color.
ROLLOVER	true, false	Sets whether rollover effects should be enabled. Rollover effects allow icons to change when the mouse is on top of a button. You must also define an IMAGE-ROLLOVER.
SELECTED	true, false	Sets the state of a button.
TEXT-HORIZONTAL	Center, Left, Right	Sets the horizontal position of text relative to an icon. Default is Left.
TEXT-VERTICAL	Center, Top, Bottom	Sets the vertical position of text relative to an icon. Default is Center.

Note: Attribute names and values are case sensitive.

UIML Language Reference

UIML Attributes

ALIGNMENT

AllowedValues	Meaning
Center, East, North, South, West	With LAYOUT:BorderLayout: Where to place an interface component in its container. Only one component may go in each region. In this usage, ALIGNMENT is specified as an attribute of the component.
LEFT, CENTER, RIGHT	With LAYOUT:FlowLayout: How to justify interface components in their container. In this usage, ALIGNMENT is specified as an attribute of the component (Frame, Panel, ScrollPane only).
CENTER, EAST, NORTH, NORTHEAST, NORTHWEST, SOUTH, SOUTHEAST, SOUTHWEST, WEST	With LAYOUT:GridBagLayout: Placement of interface component in its grid cell. In this usage, ALIGNMENT is specified as an attribute of the component.

ALIGNMENT-HORIZONTAL

AllowedValues	Meaning
Center, Left, Right	Sets the horizontal alignment of the icon and text (relative to the center of the button, not to each other). Default is Center.

ALIGNMENT-VERTICAL

AllowedValues	Meaning
Center, Top, Bottom	Sets the vertical alignment of the icon and text (relative to the center of the button, not to each other). Default is Center

AUTOSCROLLS

AllowedValues	Meaning
true, false	If true, this component will automatically scroll its contents when dragged, if contained in a component that supports scrolling.

BACKGROUND

AllowedValues	Meaning
black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set background color to color name as defined in class java.awt.Color.
One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set background color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
A non-negative integer	Set background to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

BORDER

AllowedValues	Meaning
LineBorder	Creates a single-lined border around the component. You can control the line's color and thickness with the LineBorder-Color and LineBorder-Thickness attributes.
RaisedBevelBorder	Creates a border around the component that has the effect of raising the component.
LoweredBevelBorder	Creates a border around the component that has the effect of lowering the component.
CompoundBorder	Combination of the LineBorders.
TitledBorder	Creates a border with a title. To control its appearance, use TitledBorder-Title, TitledBorder-Color, TitledBorder-Font, TitledBorder-Justification, TitledBorder-Position, and TitledBorder-Border.

UIML Attributes

BORDER-PAINTED

AllowedValues	Meaning
true, false	Sets whether the border should be painted. Default is true.

COLUMNS

AllowedValues	Meaning
A positive integer	In GridLayout: exact number of columns to use. In this usage, COLUMNS is specified as an attribute of the container.
A positive integer	In GridBagLayout: ignored.
A positive integer	With TextArea and TextField: width of the component, in characters. In this usage, COLUMNS is specified as an attribute of the component.

CONTENT

AllowedValues	Meaning
A string	Sets the initial text to display in the user interface component. In Frame: the text to appear at the top of the window. In TextArea and TextField: initial text to be displayed. In Choice and List: A sequence of items to display, separated by vertical bars.

DIRECTION

AllowedValues	Meaning
North, South, East, West	Sets the direction of the arrow on the button.

EDITABLE

AllowedValues	Meaning
true, false	Whether the text can be edited by the user. If true, the text can be edited (default).

ENABLED

AllowedValues	Meaning
true, false	If true, enables the component; otherwise, disables it. Default is true.

UIML Attributes

EXISTS

AllowedValues	Meaning
true, false	If false: <ul style="list-style-type: none">- App tag: terminate the Renderer.- RENDERING:Frame: destroys frame. If no visible frames remain, terminate Renderer If true: no effect. (Default is true.)

FILL

AllowedValues	Meaning
NONE, BOTH, HORIZONTAL, VERTICAL	Which direction the user interface component should grow in, when its container is resized and the grid cell size exceeds the user interface component size. FILL is specified as an attribute of the component.

FOCUS-PAINTED

AllowedValues	Meaning
true, false	If true, then focus is painted. (After you click on the button, a dotted line around the button shows the button has the focus. This line disappears when you click somewhere else, showing the button has lost the focus.) Default is false.

FONT_NAME

AllowedValues	Meaning
Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).

FONT_SIZE

AllowedValues	Meaning
A positive integer	Font size, in points.

FONT_STYLE

AllowedValues	Meaning
PLAIN, BOLD, ITALIC, BOLDITALIC	Font style (bold, italic, etc.)

UIML Attributes

FOREGROUND

AllowedValues	Meaning
black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set foreground color to color name as defined in class java.awt.Color.
One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set foreground color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
A non-negative integer	Set foreground to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set foreground to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none">· Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black).· A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color.· A brightness of 0 is black; a brightness of 1 is maximum color saturation.

HEIGHT

AllowedValues	Meaning
A positive integer	Interface component height, in rows.
REMAINDER	Set component height to the number of rows remaining in the grid.

HGAP

AllowedValues	Meaning
A non-negative integer	The amount of space between each component in the container and the component immediately to its left or right. (Note: This is not the amount of space around an interface component.) HGAP is specified as an attribute of the container.

IMAGE

AllowedValues	Meaning
Filename or URL	Sets the icon to display on a button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

UIML Attributes

IMAGE-DISABLED

AllowedValues	Meaning
Filename or URL	Sets the icon to display when the button is disabled. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

IMAGE-DISABLED-SELECTED

AllowedValues	Meaning
Filename or URL	Sets the icon to display when the button is disabled and selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

IMAGE-PRESSED

AllowedValues	Meaning
Filename or URL	Sets the icon to display when the button is pressed. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

IMAGE-ROLLOVER

AllowedValues	Meaning
Filename or URL	Sets the icon to display when the mouse is over the button. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

IMAGE-ROLLOVER-SELECTED

AllowedValues	Meaning
Filename or URL	Sets the icon to display when ROLLOVER effects are enabled and the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

IMAGE-SELECTED

AllowedValues	Meaning
Filename or URL	Sets the icon to display when the button is selected. Filename must point to a valid .GIF file. When specifying both path and filename, enclose the entire path and filename to the file in double-quotes.

UIML Attributes

LAYOUT

AllowedValues	Meaning
AbsoluteLayout, BorderLayout, FlowLayout, GridLayout, GridBagLayout	<p>See AWT documentation. FlowLayout is default for Panel. BorderLayout is default for Frame. AbsoluteLayout is the equivalent of a "null" LayoutManager in AWT.</p> <p>Note with AbsoluteLayout, component positions and size must be specified. Use LOCATION or LEFT and TOP to specify position; use SIZE to specify component size.</p>

LEFT

AllowedValues	Meaning
A non-negative integer.	<p>Distance, in pixels, from the left edge of the container to the left edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>

LINEBORDER-COLOR

AllowedValues	Meaning
black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Sets the color of the LineBorder's line to color name as defined in class java.awt.Color.
One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set the color of the border's line to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
A positive integer	Set the color of the border's line to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set the color of the border's line to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none"> · Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black). · A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color. · A brightness of 0 is black; a brightness of 1 is maximum color saturation.

LINEBORDER-THICKNESS

AllowedValues	Meaning
An integer	Sets the thickness of the LineBorder's line.

LOCATION

AllowedValues	Meaning
Two positive integers, separated by a comma (e.g., 125,73)	Location of the top left corner of the component, in pixels, where 0,0 is the top left corner of the container. LEFT and TOP override LOCATION. <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>

UIML Attributes

MULTIPLEMODES

AllowedValues	Meaning
true, false	Indicates whether multiple itmes in a list box may be selected. Default is false (only one item may be selected).

OPAQUE

AllowedValues	Meaning
true, false	If true, the component's background will be filled with the background color.

RENDERING

AllowedValues	Meaning
Button, Choice, Label, List, Panel, ScrollPane, TextArea, TextField	Inserts a java.awt component into a parent component. The parent component must be rendered as a Frame, Panel, or ScrollPane.
ButtonGroup	JRadioButtons nested in a ButtonGroup can be checked with mutual exclusion. The parent component must be rendered as a Frame, Panel, ScrollPane, JFrame, or JPanel.
Checkbox	Inserts a Checkbox into a parent component. The parent component must be rendered as a CheckboxGroup, Frame, Panel, or ScrollPane.
CheckboxGroup	Transforms Checkboxes into radio buttons. Place Checkboxes inside a CheckboxGroup to convert them to radio buttons. The parent component must be rendered as a Frame, Panel, or ScrollPane.
Frame	Inserts a Frame into a parent component. The parent must be an APP tag.
JButton, JCheckBox, JPanel, JRadioButton, JToggleButton	Inserts a Java Swing component into a parent component. The parent component must be rendered as a frame, panel, scrollpane, JFrame, or JPanel.
JFrame	Inserts a JFrame into a parent component. The parent must be an APP tag.
Menu	Inserts a Menu into the parent component's menubar. The Menu must be nested one or more levels within a Frame.
MenuItem	Inserts a MenuItem into a menu. The parent component must be rendered as a Menu.
None	Ignores the RENDERING tag. Useful for removing user interface elements for certain devices without having to modify the user interface definition itself (the .ui file). Note: any components contained within the component rendered as None will still be rendered.
TopPanel	Create a Panel that is not contained in any other container. Used as top level container in Active-X version of renderer.

UIML Attributes

RENDERING-PREFIX

AllowedValues	Meaning
java.awt, com.sun.java.swing	<p>Selects which windowing toolkit to use in rendering. Must be defined for every UIML tag that has a RENDERING. RENDERING-PREFIX must be defined on the APP tag.</p> <p>A convenient way to identify what toolkit to use for all user interface elements is to prefix this attribute with "+" so it applies to all components contained in the frame (e.g., +RENDERING-PREFIX:java.awt)</p>

ROLLOVER

AllowedValues	Meaning
true, false	Sets whether rollover effects should be enabled. Rollover effects allow icons to change when the mouse is on top of a button. You must also define an IMAGE-ROLLOVER.

ROWS

AllowedValues	Meaning
A positive integer	In GridLayout: exact number of rows to use. In this usage, ROWS is specified as an attribute of the container.
A positive integer	In GridBagLayout: ignored.
A positive integer	With TextArea: height of the component, in rows. In this usage, ROWS is specified as an attribute of the component.

SCROLLBARS

AllowedValues	Meaning
BOTH, HORIZONTAL_ONLY, NONE, VERTICAL_ONLY	Specifies which scrollbars should be attached to the component. Default is BOTH. Using SCROLLBARS requires explicitly defining the component's CONTENT, COLUMNS, and ROWS attributes.

SELECTED

AllowedValues	Meaning
true, false	Sets the state of a button.

UIML Attributes

SIZE

AllowedValues	Meaning
Two positive integers, separated by a comma (e.g., 125,73)	<p>Sets the interface component's size, in pixels. This attribute is implemented:</p> <ul style="list-style-type: none">- for components whose container's LAYOUT is AbsoluteLayout, and- for Frames, regardless of layout. <p>Otherwise SIZE is a preference that the layout manager may or may not elect to use.</p> <p>Note also that when using AbsoluteLayout, all components' size must be specified with the SIZE attribute.</p>

SIZE-MAXIMUM

AllowedValues	Meaning
Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's maximum size, in pixels.

SIZE-MINIMUM

AllowedValues	Meaning
Two positive integers, separated by a comma (e.g., 125,73)	Sets the interface component's minimum size, in pixels.

STATE

AllowedValues	Meaning
true, false	If true, the checkbox is checked when initially rendered.

TEXT-HORIZONTAL

AllowedValues	Meaning
Center, Left, Right	Sets the horizontal position of text relative to an icon. Default is Left.

TEXT-VERTICAL

AllowedValues	Meaning
Center, Top, Bottom	Sets the vertical position of text relative to an icon. Default is Center.

UIML Attributes

TITLED BORDER-BORDER

AllowedValues	Meaning
LineBorder, RaisedBevelBorder, LoweredBevelBorder, CompoundBorder	Specifies what line style to use for a TitledBorder.

TITLED BORDER-COLOR

AllowedValues	Meaning
black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow	Set titled-border color to color name as defined in class java.awt.Color.
One of the color names from java.awt.Color (see above) followed by a - or a + (e.g., red- or blue+).	A suffix of - means darken the color; a suffix of + means brighten the color. (These correspond to java.awt.Color.darken() and .brighten().)
0x<six hex digits> (e.g., 0xFF0000 for pure red).	Set titled-border color to RGB (red, green, blue) color specified as 0xRRBBGG, where RR denotes one of 256 intensities of red (00 to FF), BB denotes the intensity of blue, and GG denotes the intensity of green.
An integer	Set titled-border color to RGB color represented by a single integer. Bits 0-7 specify one of 256 intensities of blue; bits 8-15 green; and bits 16-23 red.
<float>,<float>,<float> where <float> is between 0 and 1 inclusive (e.g., .667,1,1 for pure blue).	Set background to HSB (hue, saturation, brightness) color specified as a triple of numbers, where the first is hue, the second is saturation, and the third is brightness. <ul style="list-style-type: none">· Hue represents the rainbow (0 is white, near 0 are red, near 1 is magenta, and 1 is black).· A saturation of 0 is either white or black, depending on brightness; a saturation of 1 is pure hue color.· A brightness of 0 is black; a brightness of 1 is maximum color saturation.

TITLED BORDER-FONT-NAME

AllowedValues	Meaning
Dialog, DialogInput, Monospaced, SansSerif, Serif	Generic name of font, which is mapped to a specific font on the platform on which Render runs (e.g., sansserif might be mapped to Helvetica).

TITLED BORDER-FONT-SIZE

AllowedValues	Meaning
A positive integer	Font size, in points.

UIML Attributes

TITLEDBORDER-FONT-STYLE

AllowedValues	Meaning
plain, bold, italic, bold italic	Font style (bold, italic, etc.)

TITLEDBORDER-JUSTIFICATION

AllowedValues	Meaning
Left, Center, Right	Sets whether the title should appear on the left, center, or right. Default is Left.

TITLEDBORDER-POSITION

AllowedValues	Meaning
Above-Top, Top, Below-Top, Above-Bottom, Bottom, Below-Bottom	Sets whether the title should appear above, on, or below the top or bottom line of the border. Default is Top.

TITLEDBORDER-TITLE

AllowedValues	Meaning
A string	Sets the title to place on the title-border.

TOOLKIT

AllowedValues	Meaning
jfc	<p>Selects which toolkit set to use in rendering. Must be defined for every UIML tag that has a RENDERING. TOOLKIT must be defined on the APP tag.</p> <p>Normally, TOOLKIT is scoped to easily define the toolkit to use for all elements in the application. This is done by prefixing TOOLKIT with "+" so it applies to all components contained in the application (e.g., +TOOLKIT:jfc).</p>

TOOLTIP

AllowedValues	Meaning
A string	Sets text to display when the mouse is over a component.

UIML Attributes

TOP

AllowedValues	Meaning
A non-negative integer	<p>Distance, in pixels, from the top edge of the container to the top edge of the component. Overrides LOCATION.</p> <p>Note this attribute measures from the outside edge of the container. So it's possible to position components so they're partially or completely hidden by the window's title bar.</p> <p>Note also that when using AbsoluteLayout, all components' position must be specified with either LOCATION or the LEFT and TOP attributes.</p>

VGAP

AllowedValues	Meaning
A non-negative integer	The amount of space between each component in the container and the component immediately above or below. (Note: This is not the amount of space around an interface component.) VGAP is specified as an attribute of the container.

VISIBLE

AllowedValues	Meaning
true, false	Specifies whether to show the user interface component (true shows; false hides). The default is true.

WIDTH

AllowedValues	Meaning
A positive integer	Interface component width, in columns.
REMAINDER	Set component width to the number of columns remaining in the grid.

XPAD

AllowedValues	Meaning
A positive integer	Amount of blank space, in pixels, added to left and right side of the interface component. XPAD is specified as an attribute of the component.

UIML Attributes

XPLACE

AllowedValues	Meaning
A positive integer	The column number in which to locate the interface component. In this usage, XPLACE is specified as an attribute of the component.
RELATIVE	Positions the component in sequence following its peers. In this usage, XPLACE is specified as an attribute of the component.

XWEIGHT

AllowedValues	Meaning
A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra vertical space as window is resized vertically; magnitude determines amount of space allocated relative to other interface component with positive XWEIGHTs.</p> <p>XWEIGHT is specified as an attribute of the component.</p>

YPAD

AllowedValues	Meaning
A positive integer	Amount of blank space, in pixels, added to above and below the interface component. YPAD is specified as an attribute of the component.

YPLACE

AllowedValues	Meaning
A positive integer	The row number in which to locate the interface component. In this usage, YPLACE is specified as an attribute of the component.
RELATIVE	Positions the component in sequence following its peers. In this usage, YPLACE is specified as an attribute of the component.

UIML Attributes

YWEIGHT

AllowedValues	Meaning
A float between 0 and 1	<p>If 0: Do not give the interface component extra space if window is resized.</p> <p>If >0: Give the interface component extra horizontal space as window is resized horizontally; magnitude determines amount of space allocated relative to other interface components with positive YWEIGHTs.</p> <p>YWEIGHT is specified as an attribute of the component.</p>