Client Applet Controls

The Client-Side Components feature of WebObjects includes many ready-made Java applets for common controls (button, text field, and so on) on a World Wide Web page. These controls are in the next.wo.client.controls package. This chapter catalogues these controls, describes their keys and actions, and tells you how to use them in a WebObjects application.

Introduction

The applets provided by NeXT "wrap" certain graphical user-interface classes in java.awt.*. They also implement the code necessary to interact with an instance of the SimpleAssociation class and thus "push" and "pull" values between the server and the client sides of an application.

A WOApplet dynamic element represents each of these applets on the server side of a WebObjects application. You must initialize this element with the location of the applet's ".class" file, the applet's dimensions, and its association class. You may also initialize any applet-specific key or action. The following is a sample set of assignments made in a declarations (".wod") file:

```
PASSWORD : WOApplet {
    code = "next.wo.client.controls.TextFieldApplet.class";
    codebase = "/WebObjects/Java";
    width = "200";
    height = "20";
    associationClass = "next.wo.client.SimpleAssociation";
    stringValue = nameString;
    echoCharacter = "*";
    action = "validateUser";
    enabled = isNotValidated; // superclass key (AWTApplet)
};
```

In this case, the stringValue and echoCharacter attributes are the keys and action is the action of TextFieldApplet. The variable "nameString" is probably declared in the component in which this dynamic element appears whereas "validateUser" is a method of the component that is invoked when the user presses Return in this text field. (Note that methods should be quoted.) The echoCharacter attribute is assigned a string constant.

Notice, however the enabled attribute. This key is not defined by TextFieldApplet, but by its superclass, AWTApplet, which is also the superclass for all NeXT-provided applets. Through inheritance, these derived applets accept values for the keys defined by AWTApplet. See "A WebObjects application downloads initial values to the applets after they're created in the browser. These initial values include constants specified in the ".wod" file and values of variables initialized in the component's init methods. Thereafter the values of applet keys are synchronized twice with the variables they are bound to in the WOApplet declaration, once before the triggering of an action method in the server and once again after the method returns nil.," below, for descriptions of these common keys

A WebObjects application downloads initial values to the applets after they're created in the browser. These initial values include constants specified in the ".wod" file and values of variables initialized in the component's init methods. Thereafter the values of applet keys are synchronized twice with the variables they are bound to in the WOApplet declaration, once before the triggering of an action method in the server and once again after the method returns nil.

Superclass Attributes

AWTApplet is the base class for the applet controls provided with WebObjects. AWTApplet has keys for common applet attributes: enabled status, foreground color, and background color. The effect of these keys depends on particular applets and particular browsers. For example, no foreground or background colors for a ButtonApplet (other than gray) are allowed. Some browsers do not support certain colors.

enabled

If enabled evaluates to "NO", the applet appears in the page but is not active. The disabled state is manifest differently for each control. For example, a TextFieldApplet will not accept the insertion of a cursor. A disabled button has a grayed-out title and doesn't react to clicks. The default setting is "YES".

foregroundColor

The color of the applet foreground. The foreground of an applet depends upon the type of object. The TextFieldApplet, for example, displays typed text in the foreground color. You must specify colors as a string. The string is either the name of a color (see "Color Names," below) or it may contain six hexadecimal digits to be interpreted in pairs as RGB values. Colors specified as hexadecimal values may start with the "#" character (although this is not required).

backgroundColor

The color of the applet background. The background of an applet depends upon the type of object. The TextFieldApplet, for example, displays the area behind typed text in the background color. You must specify colors as a string. The string is either the name of a color (see "Color Names," below) or it may contain six hexadecimal digits to be interpreted in pairs as RGB values. Colors specified as hexadecimal values may start with the "#" character (although this is not required).

Color Names

The following color names, defined as static class variables by java.awt.Color, are acceptable as string values for the foregroundColor and backgroundColor keys:

black	blue	cyan	darkGray
gray	green	lightGray	magenta
orange	pink	red	white
yellow			

A Note on Declaration Synopses

The synopses in this section include the keys and values specific to each applet as well as the specific package name for each applet class file. Ellipsis suggest that the other assignments are necessary or possible (code base or superclass keys, for instance). See the section, "Integrating Client-Side Components," in the "Java Client-Side Components" cahpter of the *WebObjects Developer's Guide* for standard WOApplet assignments, such as code base and association class.

ButtonApplet

Synopsis

WOApplet { code = “next.wo.client.control.ButtonApplet.class”
title=*aTitle*; action=*method*;...};

Description

A ButtonApplet is a control that invokes an action method in the component containing the WOApplet. The ButtonApplet class “wraps” the java.awt.Button class.

title
: The title of the button. If no title is specified, the title used is “Button”.

action
: The method to invoke when this control is clicked.

CheckboxApplet

Synopsis

WOApplet { code = “next.wo.client.control.CheckboxApplet.class”
title=*aTitle*; action=*method*; checked=”YES”|”NO”;...};

Description

A CheckboxApplet is a control that uses an image of a check box to indicate “off” and “on” (or unselected and selected) states. It can invoke an action method upon a change of state. The CheckboxApplet class “wraps” the java.awt.Checkbox class.

title
: The title of the check box.

action
: The method to invoke when the check box is clicked.

checked
: During page generation, if checked evaluates to “YES”, the check box appears in the checked state. In each subsequent synchronization point, checked reflects the state the user left the check box in and the state it’s set to by the action method.

ChoiceApplet

Synopsis

WOApplet { code = “next.wo.client.control.ChoiceApplet.class”
itemList=*anArray*; selectedItem=*itemIndex*; action=*method*;...};

Description

A ChoiceApplet displays a non-editable list of items from which the user can select one item.This control looks and behaves like a pull-down list or combo box. This class is based on the java.awt.Choice class.

itemList
: An array of strings representing the choices available.

selectedItem
: Numeric index of the item selected in itemList. Index 0 identifies the first item in the array.

action
: The method invoked when a choice is made.

ListApplet

Synopsis

WOApplet { code = “next.wo.client.control.ListApplet.class”
itemList=*arrayOfItems*; selectedItems=*arrayOfIndices*; allowsMultipleSelection=”YES”|”NO”; action=*actionMethod*;
selectionChanged=*changeMethod*;...};

Description

A ListApplet presents a list of items that allows multiple selections. Users select and deselect items in the list by clicking them. By default, this control disallows multiple selections, so you must set allowsMultipleSelection to YES to get the multiple-selection capability. The way the *actionMethod* is invoked in the component object differs for single and multiple selections. To invoke the method with a single selection, the user double-clicks the selection (this works even in multiple-selection mode). To invoke the *actionMethod* when there are multiple selections, the user must press the Return key. A ListApplet also invokes the *changeMethod* in the server-side component (if implemented) whenever a change is made in the selection.

itemList
: An array of string objects that are displayed as the choices.

selectedItems
: An array of string objects representing the numeric indices of the selected items in the list.

allowsMultipleSelection
: If this value of this key evaluates to “YES”, the user can select multiple items by clicking them in series. If “NO”, the prior selection is deselected every time the user makes a new selection.

action
: The method invoked when the user double-clicks a single item in the list or presses the Return key.

selectionChanged
: The method invoked when an item in the list is selected or deselected.

RadioGroupApplet

Synopsis

WOApplet { code = "next.wo.client.control.RadioGroupApplet.class"
iitemList=*anArray*; selectedItem=*itemIndex*; action=*method*;...};

Description

A RadioGroupApplet is a control for mutually exclusive choices. It is a vertical matrix of check boxes (see CheckboxApplet) of which only one box can be checked at a time. The RadioGroupApplet class "wraps" the java.awt.CheckboxGroup class.

iitemList
: An array of strings for the titles of the radio buttons.

selectedItem
: Numeric index of the check box selected in itemList. Index 0 identifies the first item in the array.

action
: The method invoked when a choice is made.

ScrollingTextApplet

Synopsis

WOApplet { code = “next.wo.client.control.ScrollingTextApplet.class”
stringValue=*text*;...};

Description

A ScrollingTextApplet is a bordered area on the page that allows viewing and editing of one or more lines of text. If the amount of text exceeds the display area, scrollbars enable the user to move text up and down within the display area. Pressing the Return key inserts a carriage return instead of (like TextFieldApplet) triggering an action method. The ScrollingTextApplet class “wraps” the java.awt.TextArea class.

Note: A ScrollingTextApplet does not wrap text as it approaches the right side of the display area. Instead, it scrolls text to the left. This is a limitation of java.awt.TextArea.

stringValue

During page generation, stringvalue contains the initialized value. In each subsequent synchronization point, it contains any modification to the text the user makes and the text the action method assigns to it.

TextFieldApplet

Synopsis

WOApplet { code = “next.wo.client.control.TextFieldApplet.class”
stringValue=*text*; echoCharacter=*character*; action=*method*;...};

Description

A TextFieldApplet displays a line of editable text within a rectangular border. Pressing the Return key can trigger an action method in the server-side component. This control also can function as a password field where entered characters are masked by a specified replacement character.TextFieldApplet “wraps” the java.awt.TextField class.

stringValue
: During page generation, stringvalue contains the initialized value. In each subsequent synchronization point, it contains any modification to the text the user makes and the text the action method assigns to it.

echoCharacter
: The character to display when the user types a character in the field (often an asterisk or a space character). This character is specified as a string in the “.wod” file, not as a character enclosed by single quotation marks.

action
: The method invoked when the user presses the Return key.