Adobe

Using Web Objects

Adobe® GoLive™ version 4.0

Adobe GoLive 4.0 Using Web Objects for Macintosh

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA

Adobe Systems Europe Limited, Adobe House, Edinburgh EH11 4DU, Scotland, United Kingdom

Adobe Systems Co., Ltd., Yebisu Garden Place Tower, 4-20-3 Ebisu, Shibuya-ku, Tokyo 150, Japan

Adobe Systems Pty. Ltd., P.O. Box 672, 18-20 Orion Road, Lane Cove, New South Wales 2066, Australia

Printed in the U.S.A.

Part Number: 90016252 (3/99)

Table of Contents

Using WebObjects with Adobe® GoLive™

Using WebObjects with Adobe® GoLive™

This manual describes Adobe GoLive support for WebObjects™, a powerful development environment for creating dynamic Web pages.

General

Adobe GoLive comes with an easy-to-use front end for WebObjects 3.5, the object-oriented environment from Apple Computer, Inc., for developing high-end World Wide Web applications.

Adobe GoLive makes creating dynamic content for WebObjects easy. It lets Web designers and application developers share the workload involved in producing dynamic, interactive applications for the Web. Adobe GoLive allows Web designers and application developers to work together efficiently: While the designer creates the page layout, the developer can set up the logic acting behind the scenes.

For example, Adobe GoLive lets the Web author wrap up portions of the page into *WOConditional* tags to dynamically adjust the page display to the audience's requirements. Depending on user input, the Web page will be dynamically generated by dynamic elements embedded in its HTML code. Combined with conditional logic that the application developer programs or scripts, this capability gives Web designers a powerful tool for integrating true interactivity.

What You Need to Get Started

To create interactive Web presentations using WebObjects, you need *WebObjects Enterprise*, the development environment from Apple Computer. Serving pages requires *WebObjects Server*. For more detailed information on development and deployment licenses, please consult Apple's Web sites at *http://www.apple.com/webobjects/* and *http://www.enterprise.apple.com*.

In addition, to use Adobe GoLive's WebObjects tools, the *Modules* folder in the Adobe GoLive program folder must contain the *WebObjects Module*. If you have trouble finding it, open the *Modules Manager* in Adobe GoLive's *Preferences*, locate the module in the list, then click its checkbox to enable it and relaunch Adobe GoLive. If it doesn't appear in the *Modules Manager*, you'll have to install it from the Adobe GoLive CD-ROM.

Manual Overview

This manual is subdivided in nine major sections:

- *Adobe GoLive Approach to WebObjects*, starting on page 2, lists the major components of a WebObjects application and defines Adobe GoLive's relationship with the WebObjects environment.
- *Adobe GoLive WebObjects Tools*, starting on page 3, provides an overview of the tools that the application supplies.
- *WebObjects in the Web Database*, starting on page 5, explains how the user can manage the WebObjects inventory in Adobe GoLive's *Web Database*.
- *WebObjects Types*, starting on page 12, lists the basic WebObjects tags and provides instructions for use.
- *WebObjects Client-Side Components*, starting on page 57, describes the Client-Side Components that Adobe GoLive supports and includes instructions for use.
- *WebObjects Forms*, starting on page 77, lists the *WebObjects Forms* tags and discusses how to use them.
- *WebObjects Header Tags*, starting on page 100, describes optional header tags and how to use them.
- *WebObjects Frames*, starting on page 103, describes Adobe GoLive's inventory of conditional WebObjects frames and how to insert frames and set them up.
- *WebObjects Preferences*, starting on page 107, specifies how users can customize Adobe GoLive's WebObjects editing environment.

Adobe GoLive Approach to WebObjects

When writing a WebObjects application, developers create components and connect them. A component is a page or portion of a page that contains HTML content and behavior and is located in its own directory. Components generally consist of the following files:

- An HTML template (suffix is ".html") that specifies how the page looks. This can be any HTML page built with the WebObjects editing functionality of Adobe GoLive.
- A declarations file (suffix is ".wod") that binds the dynamic elements on the HTML template page to the script's variables and actions. This file is automatically written by Adobe GoLive as the Web author adds WebObjects to the HTML template page.
- A script file (suffix is ".wos") that defines the component's attributes and implements its behavior. The application developer usually writes this script file using WebScript™, a proprietary scripting language for WebObjects.

- If necessary, any images or other resources referenced by the component.

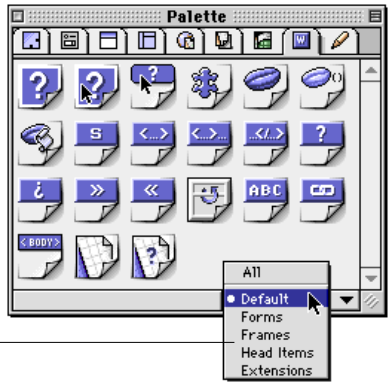
As the preceding list indicates, Adobe GoLive lets Web authors generate the "visual" part of a component and declare variables. Programming the logic is only possible in the WebObjects development environment.

Adobe GoLive WebObjects Tools

Adobe GoLive comes with a complete inventory of WebObjects tags, so designers can easily build the visual framework of dynamic HTML pages and interface smoothly with the logic that application developers program.

In five dedicated views, the *WebObjects* tab of the *Palette* contains a complete selection of building blocks that let you add full interactivity to your pages—with drag & drop ease.

The Palette and Its View Control Menu



The Palette's view control menu toggles between the various choices of WebObjects elements.

WebObjects-Specific Inspectors

You can inspect any elements placed in your page using a context-sensitive *WebObjects Inspector*. The *Inspector* window lets you set up each WebObjects element individually by choosing object-specific parameters and entering additional attributes as necessary.

WebObjects and HTML Fragments

WebObjects support requires the use of HTML fragments, allowing pages to be dynamically composed of exchangeable portions. For more information, see Chapter 4 of the *User Guide*.

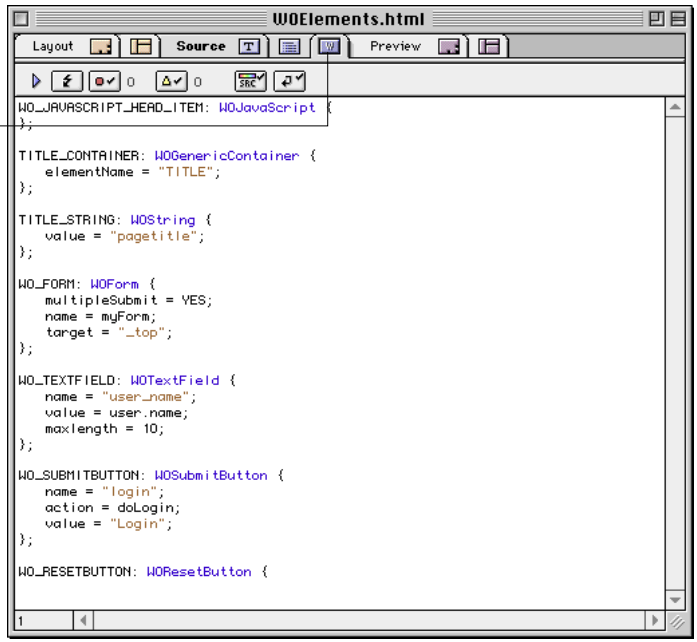
The Declaration View

Any entry to the *Inspector* window is automatically written to the declarations file (*.wod*), which you can view and edit by opening the *Declaration* tab of the main document window.

The Declaration View

Click this tab rider to open the Declaration View.

Each WebObjects element inserted in the main document window automatically generates a declaration in the declarations file.



As with the *Source* editor, the *Declaration View* supports color syntax highlighting and displays the current line number to let you keep track of your code with ease.

Two syntax highlighting control buttons above the editor text box provide convenient shortcuts for often-used commands:

Button	Function
	The *Syntax highlighting* button toggles syntax highlighting on and off. Settings in the *Preferences* dialog box control the appearance of the individual WebObjects elements (see *Color Preferences* on page 111).
	The *Soft Wrap* button toggles the wrapping of source code at the margin of the declaration view window on and off.

Syntax checking is also available to ensure 100 percent error-free code usage in the declarations file. The entries in the *WebObjects* tab of the *Web Database* supply the syntax rules.

Button	Function
	The triangle control opens and closes the error log section of the *Declaration View* window. It opens automatically when the syntax checker detects faulty code.
	The *Check Syntax* button launches syntax checking, which highlights errors and incomplete elements as it verifies the page line by line.
	When enabled, the *Display Errors* button and the error counter next to it show the number of syntax errors in the page.
	When enabled, the *Display Warnings* button and the error counter next to it show the number of faulty links in the page.

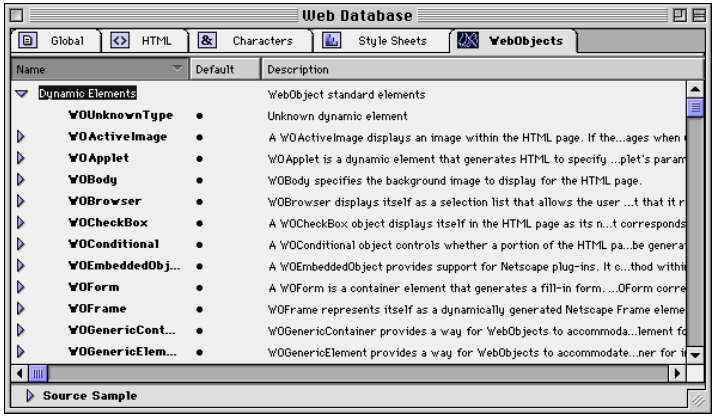
WebObjects in the Web Database

Adobe GoLive's built-in Web Database comes with a complete inventory of WebObjects elements to help Web authors choose the proper building blocks for dynamic pages.

Just as with its HTML, CSS, and Chars counterparts, the content of the *WebObjects* tab of the Web Database controls the standard options and default values that the element-specific *Inspectors offer*, thus ensuring consistent use of WebObjects elements throughout your site.

The *WebObjects* tab of the *Web Database* window appears below.

The WebObjects Tab of the Web Database Window



Editing the WebObjects Inventory in the Web Database

Editing WebObjects in the Web Database is simple and straightforward, allowing users to keep their inventory up-to-date by adding new tags as they emerge. You can add new WebObjects elements, attributes, default values, and descriptions as well as edit or delete existing entries from the list when they become obsolete. A complete set of editing tools is available in the *Special* menu and in a series of context-sensitive *Inspector* windows that pop up as you select content from the database.

Once you store an element you define in the Web Database, you can use it by inserting a generic WebObjects and choosing the proper definition in the *Inspector*. (See *WOGenericElement* on page 43, *WOGenericContainer* on page 45, and *ReusableComponent* on page 55.)

Adding, Updating, or Removing WebObjects Elements

The Web Database provides a convenient set of editing tools for adding, updating, or removing WebObjects elements and their attributes, values, and descriptions.

You can add the following items:

- new sections
- new types
- new attributes
- new enums (lists of known values)

The screenshot below shows a new *Upcoming Components* section, a new *WONew_Tag* tag, and a *new_boolean* attribute added to the *WebObjects* tab of the Web Database.

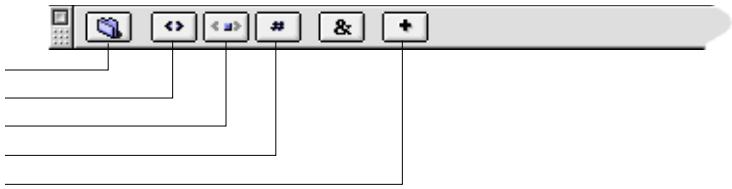
A New Section, WebObjects Tag, and Attribute Added to the Web Database

The following sections include instructions for adding new items.

Web Database Editing Tools

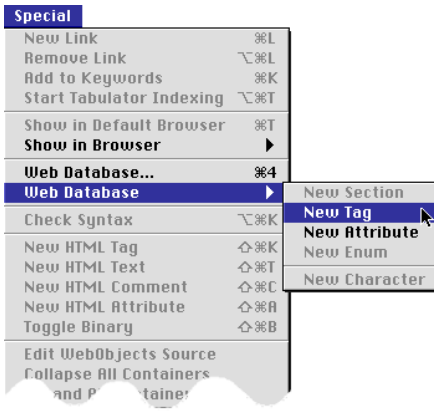
Editing tags in the Web Database is simple and straightforward. The context-sensitive toolbar turns into the *Web Database Toolbar* when the *WebObjects* tab of the *Web Database* window is activated. This toolbar provides shortcut buttons to let you add new or duplicate sections, types, attributes, and enum properties. The *Web Database Toolbar* appears below.

The Web Database Toolbar



Equivalent editing commands appear n the *Special* menu, under the second *Web Database* item. Here is the *Special* menu with its *Web Database* submenu.

The Web Database Command in the Special Menu



Do not edit any existing tags in the Web Database unless you are absolutely sure what you're doing. Serious damage to your files may result.

Adding Sections

Sections are user-defined categories that add structure to the content of the *Web Database*. Adobe GoLive comes with two default sections: The *Dynamic Elements* section contains the full suite of dynamic objects that WebObjects 3.5 supports, while the *Reusable Components* section includes interfaces for ready-to-use code elements supplied with WebObjects 3.5, such as calendars and alert panels.

To add a section, proceed as follows:

1. Choose the *Web Database…* command from the *Special* menu to open the *Web Database*.
2. Click the *WebObjects* tab rider to display the *WebObjects* tab.
3. Click the *New Section* button on the *Web Database Toolbar* or choose *New Section* from the *Web Database* submenu of the *Special* menu.
4. An entry named *new section* appears at the bottom of the *WebObjects* tab. At the same time, the *WODB Type Section* appears in the context-sensitive *Inspector* window (see below).

The WODB Section Inspector

The WODB Section Inspector lets you name the new section.

5. Place the cursor in the *Section Name* text box and enter a section name, then press the Return key to confirm your entry.
6. The section definition is now complete (see *Upcoming Components* example on page 6). You can add new tags (see below) or use drag & drop to move existing elements from the *Dynamic Objects* or *Reusable Components* subdivisions to the new section.

Adding a WebObjects Type

To add a new WebObjects type, proceed as follows:

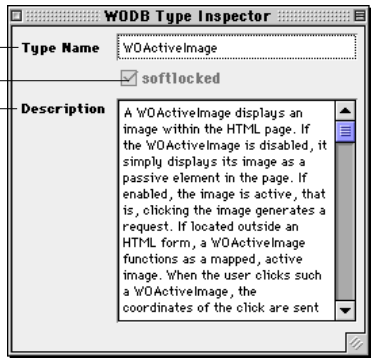
1. Select the section you want to add a new tag to.
2. Click the *New Tag* button on the *Web Database Toolbar* or choose *New Tag* from the *Web Database* submenu of the *Special* menu.
3. An entry named *new_type* appears at the bottom of the *WebObjects* tab. At the same time, the *WODB Type Inspector* appears in the context-sensitive *Inspector* window.

The WODB Type Inspector

Use this text box to name the new type.

Use this option to enable protection for user-defined types.

Use this text box to enter descriptive text.



4 Place the cursor in the *Type Name* text box and enter a name, then press the Return key to confirm your entry.

5 If desired, select the *softlocked* checkbox to protect the new tag from accidental deletion from the database.

The *softlocked* option is only available for user-defined tags. The core suite of WebObjects elements is "softlocked" by default.

6 Move the cursor to the *Description* text box and enter descriptive text.

7 The type definition is now complete (see *NEW_TAG* example on page 6). You can proceed to add attributes and enum definitions (see below).

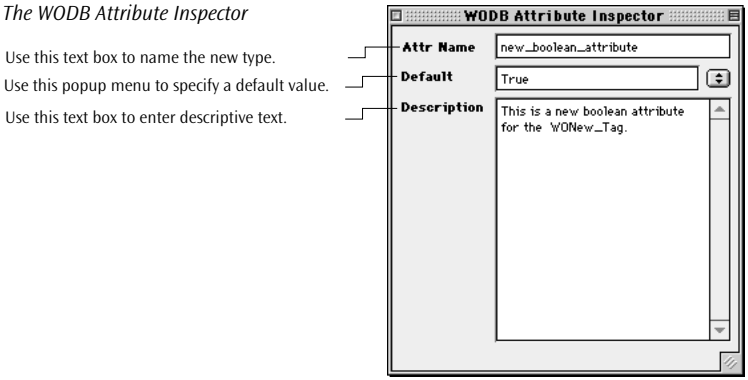
Adding a Type Attribute

To add an attribute to a WebObjects type, proceed as follows:

1 Select the type you want to add a new attribute to.

2 Click the *New Attribute* button on the *Web Database Toolbar* or choose *New Attribute* from the *Web Database* submenu of the *Special* menu.

The WODB Attribute Inspector



3 An entry named *new_attribute* appears at the bottom of the *WebObjects* tab. At the same time, the *WODB Attribute Inspector* appears in the context-sensitive *Inspector* window.

4 Place the cursor in the *Attr. Name* text box and enter an attribute name, then press the Return key to confirm your entry.

5 Move the cursor to the *Default* text box and enter a default value. If the attribute supports several known values, go to the next section and enter enum properties, then return to the *WODB Attribute Inspector* and choose one from the popup menu next to the *Default* text box.

6 Move the cursor to the *Description* text box and describe the attribute.

7 Now that the attribute definition is complete (see *new_boolean_attribute* example on page 6), you can proceed to add enum definitions (see below).

Adding Enum Definitions

Enum definitions allow for passing a suite of initial values to an object—for example, user-selectable items for a menu in a fill-in form.

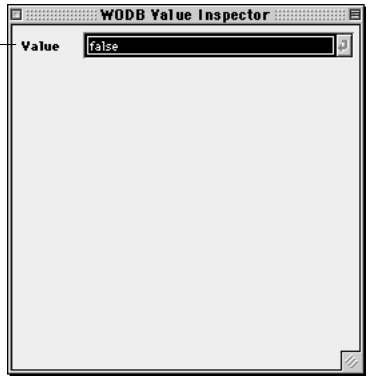
To add an optional enum option to a WebObjects type attribute, proceed as follows:

1 Select the type attribute you want to add a new enum definition to.

2 Click the *New Enum* button on the *Web Database Toolbar* or choose *New Enum* from the *Web Database* submenu of the *Special* menu.

3 An entry named *new_value* appears at the bottom of the *WebObjects* tab. At the same time, the *WODB Value Inspector* appears in the context-sensitive *Inspector* window.

The WODB Value Inspector



Use this text box to enter a value for the enum definition.

4 Place the cursor in the *Value* text box and enter an enum value, then press the Return key to confirm your entry. Repeat this step until you have specified all values that the new attribute supports.

The enum definition is now complete.

Updating Entries

To update a WebObjects element, proceed as follows:

1 Select the WebObjects element to be changed in the *WebObjects* tab of the *Web Database* window.

2 Make the desired changes by editing the name, description, or other item of your choice, then press the Return key to have your changes written to the database.

Duplicating Items

To duplicate a WebObjects element, proceed as follows:

1 Select the WebObjects element to be duplicated in the *WebObjects* tab of the *Web Database* window.

2 Click the *Duplicate Item* button on the *Web Database Toolbar*.

3 An entry with named *itemname_copy* appears at the bottom of the *WebObjects* tab. At the same time, the respective *Inspector* appears in the context-sensitive *Inspector* window.

4 Proceed to change the copied item.

Deleting Items

To delete a WebObjects element, proceed as follows:

1. Select the WebObjects element to be deleted in the *WebObjects* tab of the *Web Database* window.
2. Press the Backspace key.

WebObjects Types

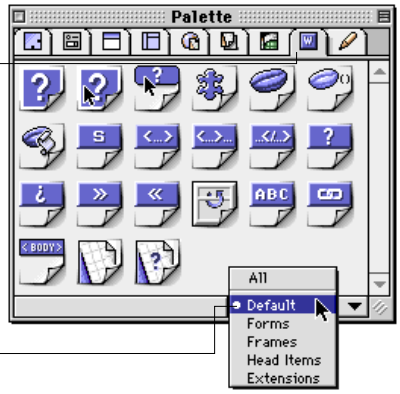
Adobe GoLive's default *WebObjects* tag inventory contains the standard suite of elements you need to build dynamic HTML pages, including visible items, such as images, applets and tables, as well as invisible items, such as state storage and repetition elements.

Displaying Default WebObjects Types

The default *WebObjects* tag inventory resides in the *Default* view of the *WebObjects* tab. To display this view, select the *Default* item from the view control menu at the bottom of the *Palette*.

The WebObjects Tab of the Palette—Default View Selected

To view the default WebObjects tag inventory, open the WebObjects tab of the Palette, …



… then select Default from the Palette's view control menu.

The following section lists the default WebObjects elements sorted in the order of their appearance:

- *WOImage* (see page 13)
- *WOActiveImage* (see page 16)
- *WOImageButton* (see page 21)
- *WOApplet* (see page 24)
- *WOParam* (see page 27)
- *WOJavaScript* (see page 29)
- *WOEmbeddedObject* (see page 30)
- *WORepetition* (see page 33)
- *WOConditional* (see page 36)

- *WOString* (see page 38)
- *WOStateStorage* (see page 41)
- *WOGenericElement* (see page 43)
- *WOGenericContainer* (see page 45)
- *WOHyperlink* (see page 48)
- *WOBody* (see page 50)
- *WOSwitchComponent* (see page 53)
- *ReusableComponent* (see page 55)
- *Table* (see page 57)

WOImage

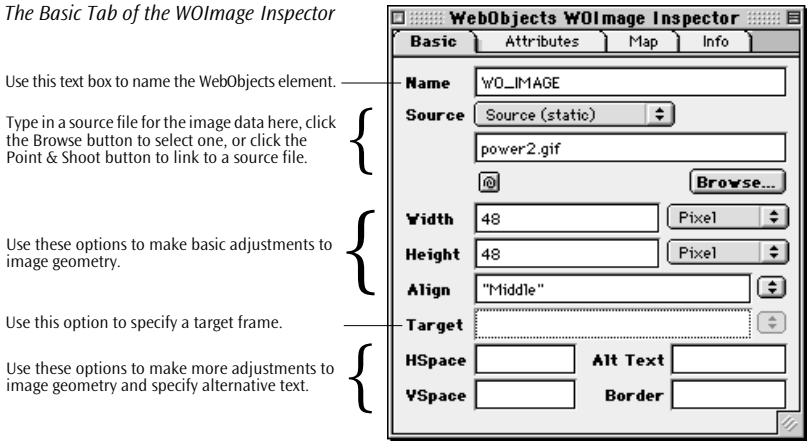
The *WOImage* icon inserts an image placeholder that can be linked with an image. As with its HTML counterpart, a *WOImage* object displays a passive, non-clickable image on the Web page.

Inserting a WOImage Placeholder

To insert a *WOImage* placeholder, proceed as follows:

1. Drag the *WOImage* icon from the *Palette* and drop it in your layout grid or document window.
2. Set up the image in the context-sensitive *Inspector* window, now titled *WOImage Inspector*. It has four tabs:
   - *Basic* lets you set the source file and geometry for the WebObjects image.
   - *Attributes* lets you inspect current attributes and add new ones.
   - *Map* can be opened, but its options are disabled. Image maps are only available for *WOActiveImage* and *WOImageButton* objects.
   - *Info* briefly describes the WebObjects element.

The Basic Tab of the WOImage Inspector



Set the following options in the *Basic* tab of the *WOImage Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *File* text box and checkbox combination to type in a path to the file containing the image data. The source file can be statically specified in the declarations file or it can be dynamically specified using an NSString, an object that responds to a description message by returning an NSString, or a method that returns an NSString. For information on the NSString class, please refer to the documentation on foundation classes, available from Apple Computer, Inc. (formerly from NeXT Software, Inc.).

Alternatively, click the *Browse* button and select an image in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

Adjusting WOImage Alignment

To adjust the alignment of an image placed in the flow of HTML code or on a layout grid, proceed as follows:

To align the image relative to text on the same line, select an option from the *Align* popup menu:

- The *Default* option uses the alignment settings of the surrounding text.
- The *Top* option aligns surrounding text with the top of the image.

- The *Middle* option horizontally centers the baseline of surrounding text with the image.
- The *Bottom* option is the default setting. It aligns the baseline of surrounding text with the bottom of the image.
- The *Left* option aligns the image to the left of the text.
- The *Right* option aligns the image to the right of the text.
- The *Texttop* option aligns the image with the top of the surrounding text.
- The *Absmiddle* option horizontally aligns the absolute center of surrounding text with the image.
- The *Baseline* option aligns the image with the baseline of the surrounding text.
- The *Absbottom* option aligns the absolute bottom of the surrounding text with the bottom of the image.

Adjusting Vertical and Horizontal Spacing

To adjust the vertical and horizontal spacing between the image and surrounding text, proceed as follows:

1. In the *HSpace* text box, type the horizontal spacing in pixels and press the Return key to confirm your entry.
2. In the *VSpace* text box, type the vertical spacing in pixels and press the Return key to confirm your entry.

Entering Alternative Text

To enter alternative text you want the browser to display instead of the image, proceed as follows:

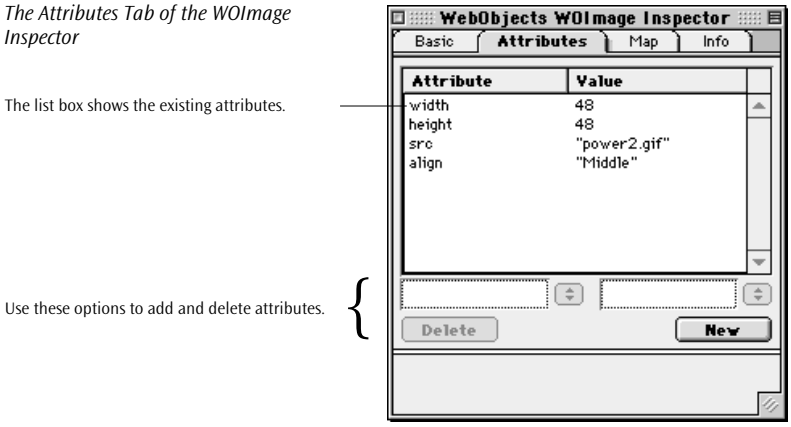
Click to place the cursor in the *Alt Text* text box, enter the desired text string, and press the Return key to confirm your entry.

Adjusting Border Width

To activate a bounding box around the WOImage and adjust its width, proceed as follows:

1. Type in the desired border width in pixels and press the Return key to confirm your entry.
2. Click the *Preview* tab to check the appearance of your image.

The Attributes Tab of the WOImage Inspector



The *Attributes* tab of the *WOImage Inspector* lets you inspect the settings made in the *Basic* tab and add new attributes as required.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOImage*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press the Return key to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to delete a selection from the *Attributes* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

WOActiveImage

The *WOActiveImage* icon inserts an image placeholder that displays an image within the HTML page. If the WOActiveImage is disabled, it simply displays as a passive element in the page. If enabled, the image is active—that is, clicking the image generates a request.

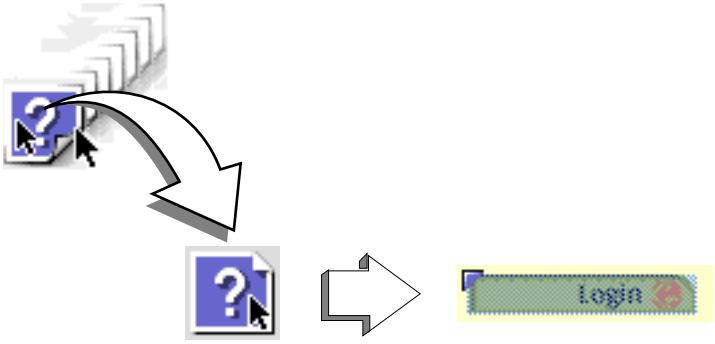
If located outside an HTML form, a WOActiveImage functions as a mapped, active image. When the site visitor clicks such a WOActiveImage, the coordinates of the click are sent back to the server. Depending on where the site visitor clicks, different actions can be invoked. An image map file associates actions with each of the defined areas of the image.

Within an HTML form, a WOActiveImage acts as a graphical submit button. You typically use WOActiveImage when you need more than one submit button within a form.

To insert a *WOActiveImage* placeholder, proceed as follows:

1 Drag the *WOActiveImage* icon from the *Palette* and drop it in your layout grid or document window.

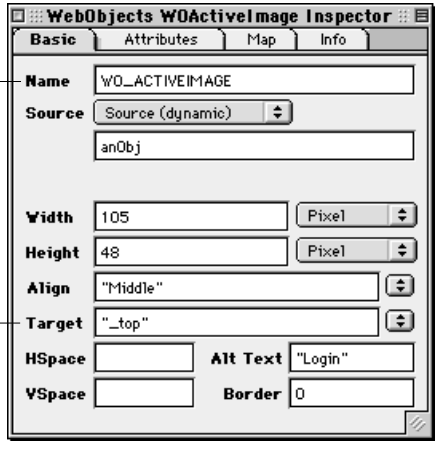
Inserting a WOActiveImage Placeholder



2 Set up the image in the context-sensitive *Inspector* window, now titled *WOActiveImage Inspector*. It has four tabs:

- *Basic* lets you set the source file and geometry for the WebObjects image.
- *Attributes* lets you inspect current attributes and add new ones.
- *Map* allows you to create server-side clickable image maps.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOActiveImage Inspector



Use this text box to name the WebObjects element.

Type in a source file for the image data here, click the Browse button to select one, or click the Point & Shoot button to link to a source file.

Use these options to make basic adjustments to image geometry.

Use this option to specify a target frame.

Use these options to make more adjustments to image geometry and specify alternative text.

Set the following options in the *Basic* tab of the *WOActiveImage Inspector*:

- The *Name* text box displays the name of the current WebObjects element.
- Use the *File* text box and checkbox combination to type in a path to the file containing the image data. The source file can be statically specified in the declarations file or it can be dynamically specified using an NSString, an object that responds to a description message by returning an NSString, or a method that returns an NSString. For information on the NSString class, please refer to the documentation on foundation classes, available from Apple Computer, Inc. (formerly from NeXT Software Inc.).

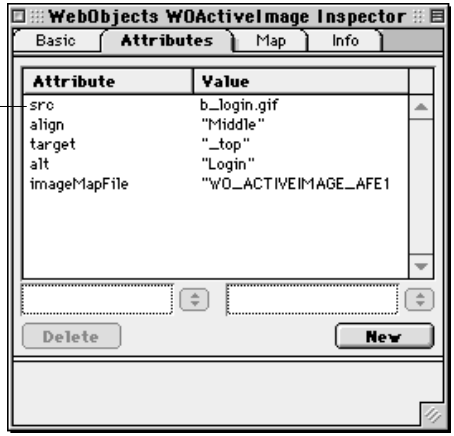
Alternatively, click the *Browse* button and select an image in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- Use the *Target* popup menu to choose a frame in a frame set that will receive the page returned as a result of the site visitor's click.

For instructions on how to adjust image geometry, specify a target frame, set up display properties, and enter alternative text, please refer to the respective instructions for the *WOImage* object:

- *Adjusting WOImage Alignment* on page 14
- *Adjusting Vertical and Horizontal Spacing* on page 15
- *Entering Alternative Text* on page 15
- *Adjusting Border Width* on page 15

The Attributes Tab of the WOActiveImage Inspector



The *Attributes* tab of the *WOActiveImage Inspector* lets you inspect the settings you made in the *Basic* and *Map* tabs and add new attributes as necessary.

To add and delete attributes, proceed as follows:

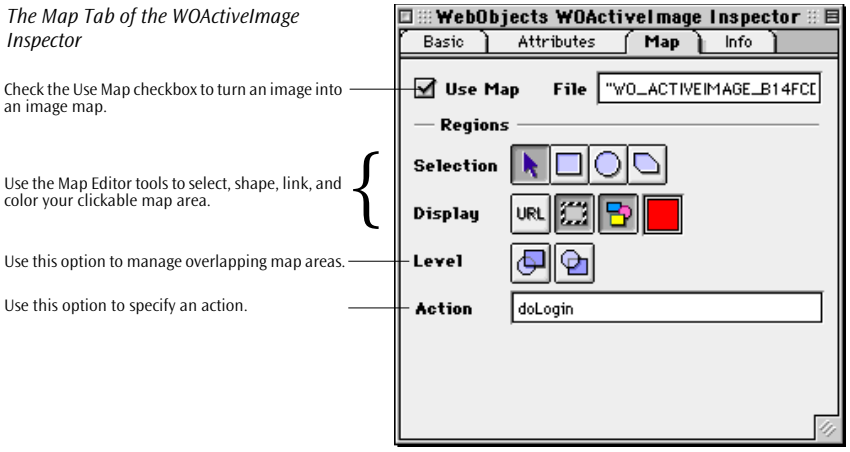
1. Click the *New* button to add new attributes for the WOActiveImage. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

The *Map* tab of the *WOActiveImage Inspector* lets you edit server-side clickable image maps served by the WebObjects runtime environment.

Adobe GoLive provides a complete selection of easy-to-use drawing and selection tools that lets you create clickable maps, edit them right on top of the image, and link them to the WebObjects application. The program also links the coordinates and shapes of clickable maps with associated actions, automatically writing the resulting definitions to an image map file that goes to the WebObjects development environment for further processing.

The Map Tab of the WOActiveImage Inspector



To function as active images, *WOActiveImage* objects must be located outside an HTML form. They cannot be enclosed in *WOForm* tags (see page 78).

Please note that the *Use Map* checkbox is deselected when the user opens the HTML page via an FTP connection.

Because WOActiveImages are images with clickable hot spots on top of them, you will have to insert an image before you can get started.

To insert an image and convert it into a clickable map, proceed as follows:

1. Drop the *WOActiveImage* icon onto your document window.
2. With the *WOActiveImage* selected, click the *Map* tab in the *WOActiveImage Inspector* window.
3. In the *Map* tab, check the *Use Map* checkbox.
4. Adobe GoLive will create an image map file. The file name, consisting of the WebObjects element name, a hexadecimal identifier, and the suffix.*map*, will appear in the *File* text box of the *Map* tab.
5. Place the cursor in the *Action* text box and type in the name of a method or script specifying the action to occur when the site visitor clicks. Press the Return key to confirm your entry.
6. You can now edit the hot-spot area of the image map using the toolbar in the *Map* tab. The toolbar features map-specific buttons that let you shape, color, and otherwise edit your map area.

To draw a hot-spot area and adjust display options, proceed as follows:

1. Click to select one of the following map drawing tools:

This button allows you to draw rectangular hot spots.

This button lets you draw circular hot spots.

This button allows you to draw polygonal hot spots.

2. Draw the map at the desired location.

3 Click the following buttons to better identify multiple maps:


This button toggles the border around the map on and off.


This button toggles the fill pattern box on and off.


This button opens the *Color Palette*, allowing you to select a different fill color. (Red is the default color.)

This button toggles the name of the referenced resource on and off.

4 If you are working with multiple overlapping hot spots, use the following buttons to manage them:


This button activates the cursor, allowing you to resize and rearrange maps.


This button brings the current map to the front.


This button sends the current map to the back.

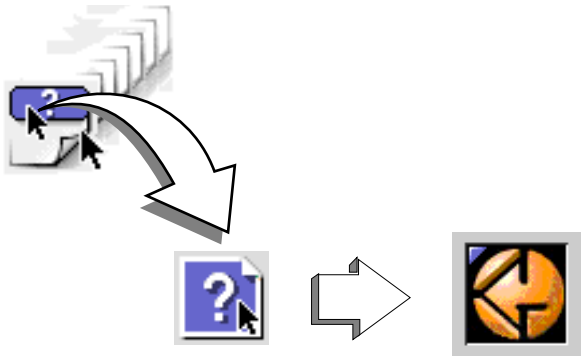
WOImageButton

WOImageButton is a graphical submit button. Clicking the image generates a request and submits the enclosing form's values. You often use WOImageButton when you need more than one submit button within a form.

To insert a *WOImageButton* placeholder, proceed as follows:

1 Drag the *WOImageButton* icon from the *Palette* and drop it in your layout grid or document window.

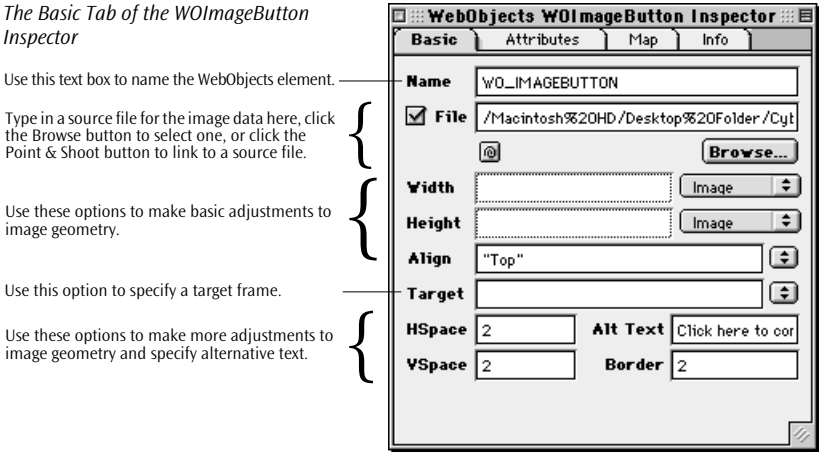
Inserting a WOImageButton



2 Set up the image in the context-sensitive *Inspector* window, now titled *WOImageButton Inspector*. It has four tabs:

- *Basic* lets you set the source file and geometry for the WebObjects image.
- *Attributes* lets you inspect current attributes and add new ones.
- *Map* allows you to create server-side clickable image maps.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOImageButton Inspector



Set the following options in the *Basic* tab of the *WOImageButton Inspector*:

- The *Name* text box displays the name of the current WebObjects element.
- Use the *File* text box and checkbox combination to type in a path to the file containing the image data. The source file can be stat-

ically specified in the declarations file or it can be dynamically specified using an NSString, an object that responds to a description message by returning an NSString, or a method that returns an NSString. For information on the NSString class, please refer to the documentation on foundation classes, available from Apple Computer, Inc. (formerly from NeXT Software Inc.).

Alternatively, click the *Browse* button and select an image in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- Use the *Target* popup menu to choose a frame in a frame set that will receive the page returned as a result of the site visitor's click.

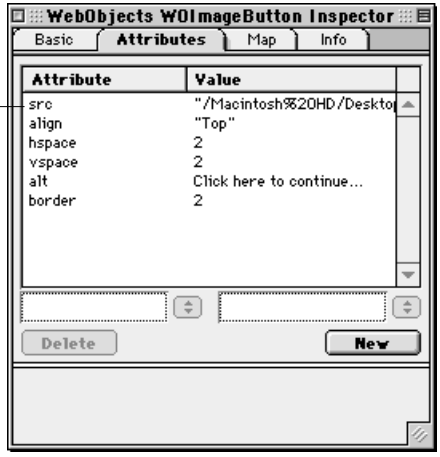
For instructions on how to adjust image geometry, specify a target frame, set up display properties and enter alternative text, please refer to the appropriate instructions for the *WOImage* object:

- *Adjusting WOImage Alignment* on page 14
- *Adjusting Vertical and Horizontal Spacing* on page 15
- *Entering Alternative Text* on page 15
- *Adjusting Border Width* on page 15

The Attributes Tab of the WOImageButton Inspector

The list box shows the existing attributes.

Use these options to add and delete attributes.



The *Attributes* tab of the *WOImageButton Inspector* lets you inspect the settings made in the *Basic* and *Map* tabs and add new attributes as required.

To add and delete attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOImageButton*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.

2 Click the *Delete* button to delete a selection from the *Attributes* list box.

For information on the options in the *Map* tab of the *WOImageButton Inspector*, please refer to the instructions on page 19 in the preceding section on the *WOActiveImage* tag.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

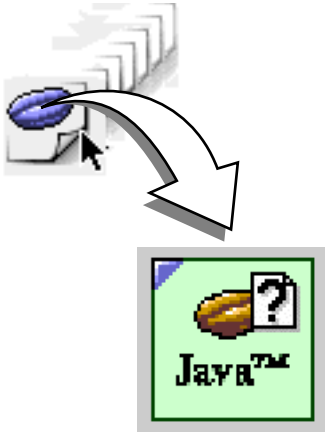
WOApplet

The *WOApplet* icon inserts a dynamic element that generates HTML to specify a Java applet. One or more *WOParam* elements pass the applet's parameters.

To insert a *WOApplet* placeholder, proceed as follows:

1 Drag the *WOApplet* icon from the *Palette* and drop it in your layout grid or document window.

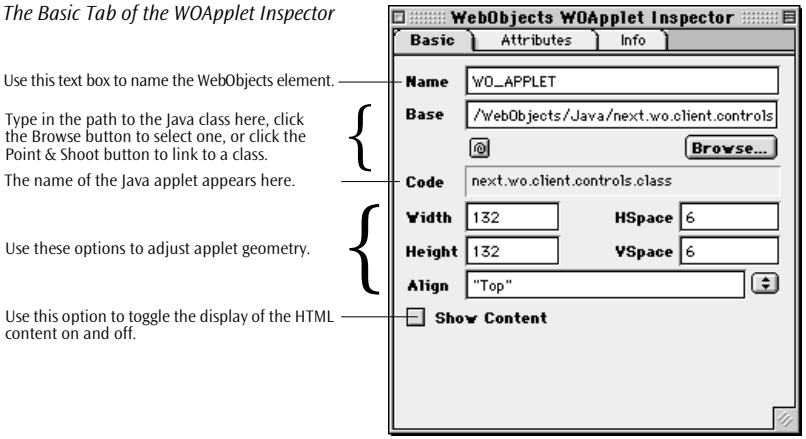
Inserting a WOApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WOApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



Set the following options in the *Basic* tab of the *WOApplet Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Data* text box to type in the path to the Java applet. If the directory containing the applet code is omitted, the applet code is assumed to be in the same directory as the template HTML file.

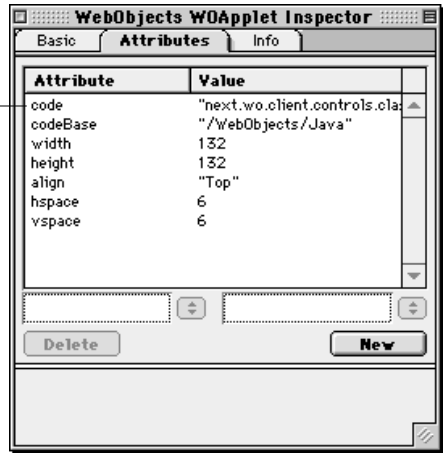
Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. The *Align* popup menu lets you select the following options:

- The *Default* option uses the alignment settings of the surrounding text.
- The *Top* option aligns surrounding text with the top of the applet.
- The *Middle* option horizontally centers the baseline of surrounding text with the applet.
- The *Bottom* option is the default setting. It aligns the baseline of surrounding text with the bottom of the applet.
- The *Left* option aligns the applet to the left of the text.
- The *Right* option aligns the applet to the right of the text.
- The *Texttop* option aligns the applet with the top of the surrounding text.
- The *Absmiddle* option horizontally aligns the absolute center of surrounding text with the applet.
- The *Baseline* option aligns the applet with the baseline of the surrounding text.
- The *Absbottom* option aligns the absolute bottom of the surrounding text with the bottom of the applet.

• Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings made in the *Basic* tab and add new attributes as required.

To add and delete attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list

box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.

2 Click the *Delete* button to remove a selection from the *Attributes* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

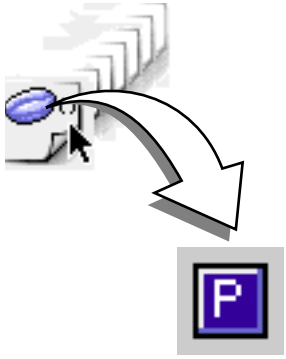
WOParam

The *WOParam* icon inserts an element that contains parameter definitions for the preceding applet. *WOParam* elements can be cascaded to insert more complex parameter definitions.

To insert a *WOParam* parameter definition element, proceed as follows:

1 Drag the *WOParam* icon from the *Palette* and drop it on the *WOApplet* placeholder on your layout grid or in the document window.

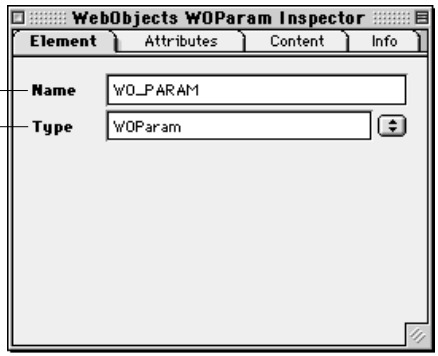
Inserting a WOParam Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOParam Inspector*. It has four tabs:
   - *Element* lets you name the WebObjects element and change its type.
   - *Attributes* allows you to add new attributes.
   - *Content* lets you view alternative HTML content, if any.
   - *Info* briefly describes the WebObjects element.

3 Make sure you have checked the *Show Content* checkbox in the *Basic* tab of the *WOApplet Inspector* (see page 26 for instructions) to display the parameters.

The Element Tab of the WOParam Inspector



Use this text box to name the WebObjects element.

Use this popup menu as a shortcut to changing the type of the object.

Set the following options in the *Element* tab of the *WOParam Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This identifies the *WOParam* object as a unique entity in case more parameter definitions follow.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if required.

The Attributes Tab of the WOParam Inspector

The list box shows the existing attributes.

Use these options to add and delete attributes.

The *Attributes* tab of the *WOParam Inspector* lets you add attributes for the preceding Java applet.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes to the parameter definition. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.

2 Click the *Delete* button to delete a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

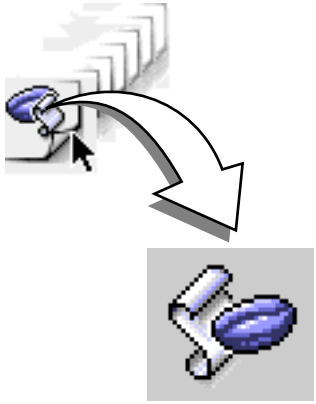
WOJavaScript

The *WOJavaScript* icon inserts an element that lets you embed a script written in JavaScript in a dynamically generated page.

To insert a *WOJavaScript* element, proceed as follows:

1 Drag the *WOJavaScript* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOJavaScript Placeholder



2 Set up the script in the context-sensitive *Inspector* window, now titled *WOJavaScript Inspector*.

The WOJavaScript Inspector

3 Set the following options in the *WOJavaScript Inspector*:

- Use the *Elementname* text box to give the script element a unique name. This identifies the script more clearly in case there are more scripts on the same page.
- Use the *File* text box to specify the path to the file containing the script. The path can be statically specified in the declaration file or it can be an NSString, an object that responds to a description message by returning an NSString, or a method that returns an NSString. For information on the NSString class, please refer to the documentation on foundation classes, available from Apple Computer, Inc. (formerly NeXT Software Inc.).
- Use the *String* checkbox and *Edit* button to embed a string containing the script in the HTML page. Clicking the *Edit* button opens Adobe GoLive's JavaScript editor, which the Adobe GoLive *User Guide* describes in more detail. Typically, ScriptString is an NSString object (see above), an object that responds to a description message by returning an NSString, or a method that returns an NSString.
- Use the *Source* text box to specify a URL for the location of the script.
- Use the *Use Comment* text box and popup menu to determine whether the script will be enclosed in an HTML comment. Setting the *Comment* option to YES will enclose the script in HTML comment tags. Scripts can generate errors in some older browsers that weren't designed to execute them, so you may want to enclose your script in an HTML comment. Browsers designed to run these scripts will still be able to execute them despite the surrounding comment tags.

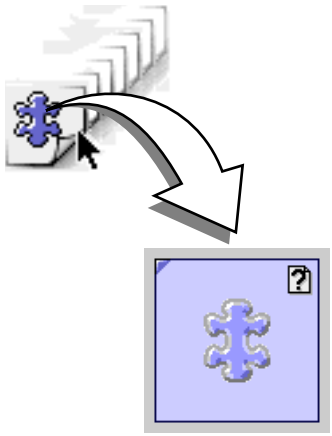
WOEmbeddedObject

The *WOEmbeddedObject* icon inserts an element that provides support for Netscape plugins.

To insert a *WOEmbeddedObject* placeholder, proceed as follows:

1 Drag the *WOEmbeddedObject* icon from the *Palette* and drop it in your layout grid or document window.

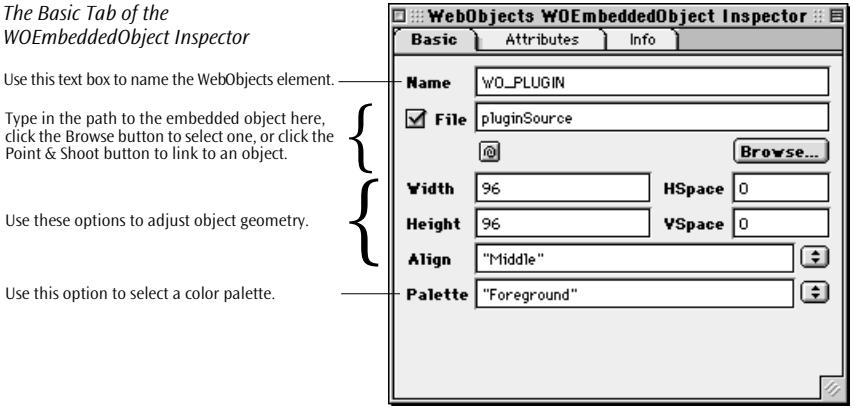
Inserting a WOEmbeddedObject Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOEmbeddedObject Inspector*. It has three tabs:

- *Basic* lets you make basic settings for *WOEmbeddedObject*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOEmbeddedObject Inspector



Set the following options in the *Basic* tab of the *WOEmbeddedObject Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name.
- Use the *File* text box and checkbox combination to type in the path to the plugin file if the embedded object's content comes from outside the WebObjects application. If the embedded

object's content is returned by a method within the WebObjects application, use the *Attributes* tab to specify the *Value* attribute.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the plugin. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the plugin. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the plugin relative to text on the same line. The *Align* popup menu lets you select one of the following options:
  - The *Default* option uses the alignment settings of the surrounding text.
  - The *Top* option aligns surrounding text with the top of the plugin.
  - The *Middle* option horizontally centers the baseline of surrounding text with the plugin.
  - The *Bottom* option is the default setting. It aligns the baseline of surrounding text with the bottom of the plugin.
  - The *Left* option aligns the plugin to the left of the text.
  - The *Right* option aligns the plugin to the right of the text.
  - The *Texttop* option aligns the plugin with the top of the surrounding text.
  - The *Absmiddle* option horizontally aligns the absolute center of surrounding text with the plugin.
  - The *Baseline* option aligns the plugin with the baseline of the surrounding text.
  - The *Absbottom* option aligns the absolute bottom of the surrounding text with the bottom of the plugin.

- Use the *Palette* text box and popup menu to select the background or foreground palette for the plugin.

The Attributes Tab of the WOEmbeddedObject Inspector

The *Attributes* tab of the *WOEmbeddedObject Inspector* lets you inspect the settings made in the *Basic* tab and add new attributes as required.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOEmbeddedObject*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press the Return key to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

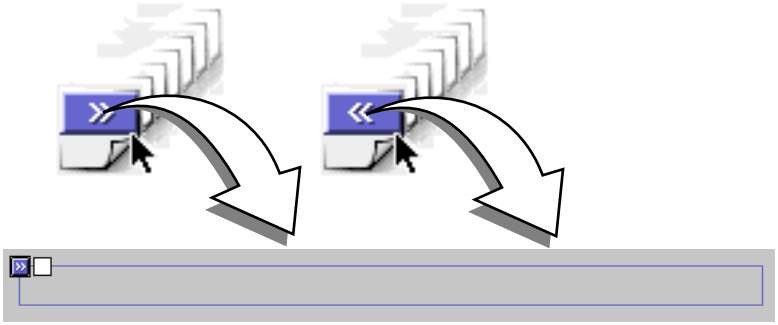
WORepetition

The *WORepetition* icons insert a container element that repeats its contents a given number of times. You can use *WORepetition* to create dynamically generated ordered and unordered lists or banks of checkboxes or radio buttons.

To insert a *WORepetition* tag, proceed as follows:

1. Drag the *WORepetition (Begin)* tag icon from the *Palette* and drop it on your layout grid or in the document window.
2. Drag the item(s) to be repeated into the box-shaped container.

Inserting a WORepetition Tag



OR

Tip: The *WORepetition* element has two display options. You can change its appearance in the *WebObjects Preferences* (see page 107). If you do so, the default box-shaped container will be replaced by the two icons shown above, which must enclose the item(s) to be repeated. An example appears on page 108.

3 Select the *WORepetition* container or the *WORepetition (Begin)* tag to set up the repetition object in the context-sensitive *Inspector* window, now titled *WORepetition Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

The Element Tab of the WORepetition Inspector

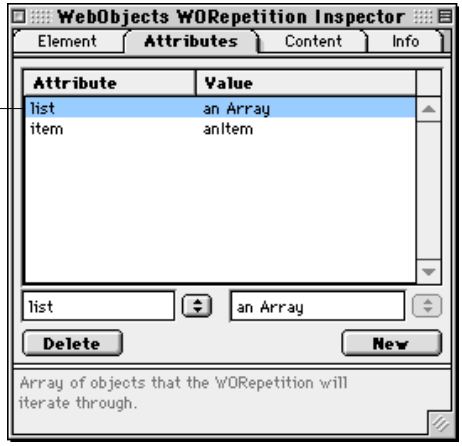
Use this text box to name the WebObjects element.

Use this popup menu as a shortcut to changing the type of the object.

Set the following options in the *Element* tab of the *WORepetition Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This identifies the *WORepetition* object as a unique entity in case more repetitions follow.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

The Attributes Tab of the WORepetition Inspector



The *Attributes* tab of the *WORepetition Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as necessary.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WORepetition* object. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Currently supported attributes include:
   - *list*: An array of objects that the WORepetition will iterate through
   - *item:* A current item in the list array
   - *index:* An index of the current iteration of the WORepetition
   - *identifier:* A value that uniquely identifies the item in the list array. Typically it is the primary key of an enterprise object
   - *count:* The number of times this element will repeat its contents

3 Click the *Delete* button to remove a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

WOConditional

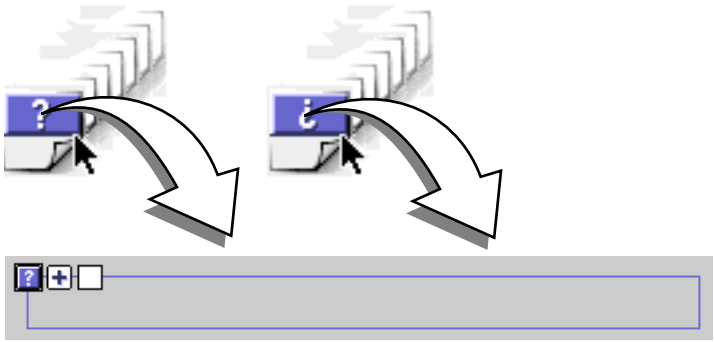
The *WOConditional* icons insert a conditional container object that controls whether a portion of the HTML page will be generated, based on the evaluation of its assigned condition.

To insert a *WOConditional* tag, proceed as follows:

1 Drag the *WOConditional (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window.

2 Drag the item(s) you want to be conditional into the box-shaped container.

Inserting a WOConditional Tag



OR

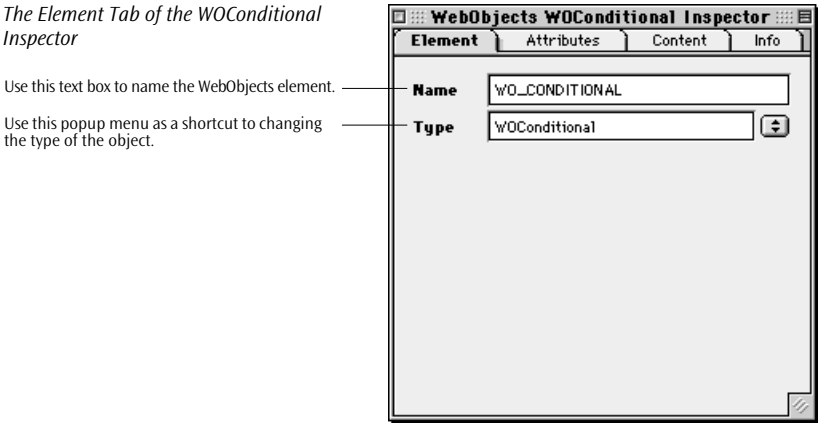
Tip: The *WOConditional* element has two display options. You can change its appearance in the *WebObjects Preferences* (see page 107). If you do so, the default box-shaped container will be replaced by the two icons shown above, which must enclose the item(s) you want to be conditional. See page 108 for an example.

3 Select the *WOConditional* container or *WOConditional (Begin)* tag to set up the conditional object in the context-sensitive *Inspector* window, now titled *WOConditional Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.

- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

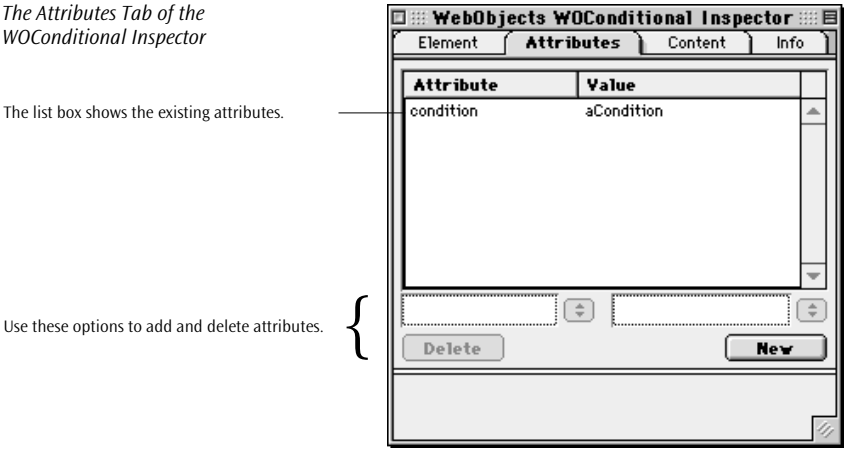
The Element Tab of the WOConditional Inspector



Set the following options in the *Element* tab of the *WOConditional Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOConditional* object as a unique entity in case more conditional portions follow in the HTML template page.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

The Attributes Tab of the WOConditional Inspector



The *Attributes* tab of the *WOConditional Inspector* lets you add new attributes and values, as required.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOConditional* object. Clicking the N*ew b*utton enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.

   Currently supported attributes include:

   - *condition*: An expression to evaluate. If the expression evaluates to YES (assuming negate is NO), the HTML code that the WOConditional object controls is emitted; otherwise it is not.
   - *negate*: Inverts the sense of the condition. By default, negate is assumed to be NO.
2. Click the *Delete* button to delete a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

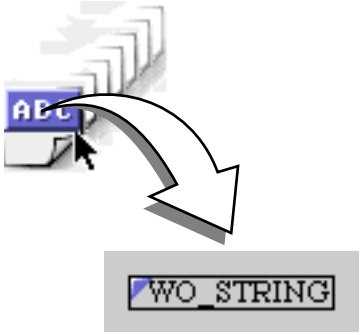
WOString

The *WOString* icon inserts an object that represents itself in the HTML page as a dynamically generated string.

To insert a *WOString* tag, proceed as follows:

1. Drag the *WOString* tag icon from the *Palette* and drop it in your layout grid or document window.

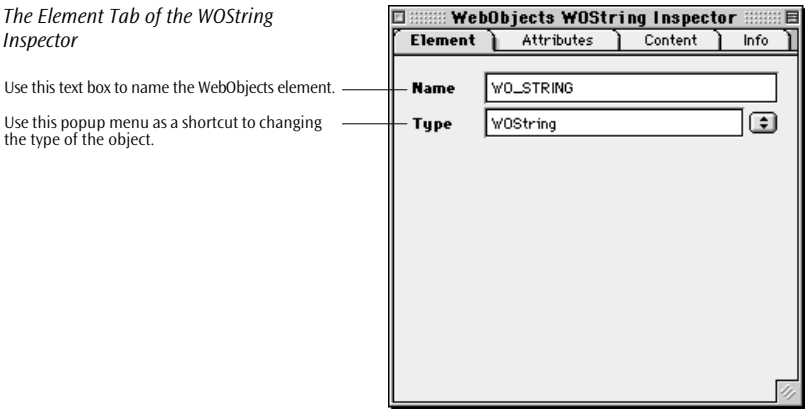
Inserting a WOString Tag



2. Select the *WOString* tag to set up the string object in the context-sensitive *Inspector* window, now titled *WOString Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

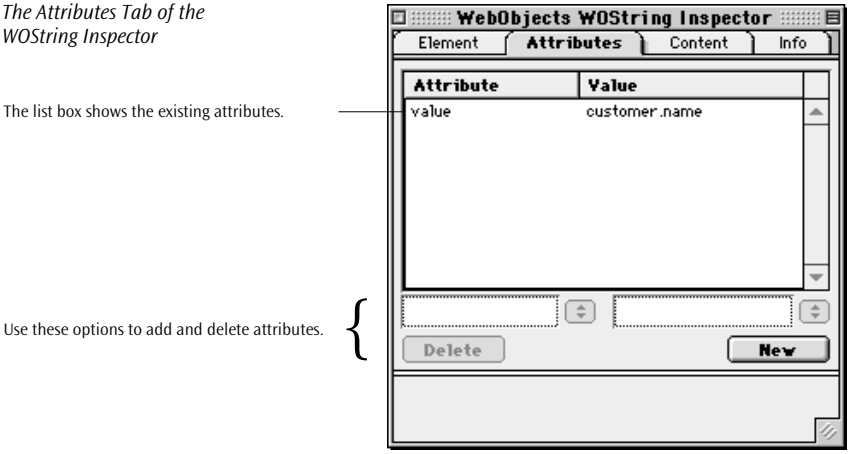
The Element Tab of the WOString Inspector



Set the following options in the *Element* tab of the *WOString Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOString* object as a unique entity in case more string objects follow in the HTML template page.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

The Attributes Tab of the WOString Inspector



The *Attributes* tab of the *WOString Inspector* lets you add new attributes and values, as needed.

To add and delete attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOString* object. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.

Currently supported attributes include:

- *value*: This attribute specifies the text to be displayed in the HTML page. *value* is typically assigned an NSString object, an object that responds to a description message by returning an NSString, or a method that returns an NSString. The NSString's contents are substituted into the HTML in the place that this dynamic element occupies. For a description of the NSString class, please refer to the documentation on foundation classes, available from Apple Computer, Inc. (formerly from NeXT Software, Inc.).
- *dateformat*: This format string specifies how to format *value* as a date. If a date format is used, *value* must be assigned an NSCalendarDate object. If *value* can't be interpreted according to the format you specify, *value* is set to nil. See the NSCalendarDate class specification in the documentation on foundation classes for a description of the date format syntax.
- *numberformat:* This format string specifies how to format *value* as a number. If a number format is used, *value* must be assigned an NSDecimalNumber object. If the element's value can't be interpreted according to the format you specify, *value* is set to nil. See the NSNumberFormatter class specification in the documentation on foundation classes for a description of the number format syntax.
- *escapeHTML*: If *escapeHTML* is set to YES, HTML tags in WOString's contents are protected from being interpreted by the browser; otherwise, they are not. By default, WebObjects tries to ensure that the contents of a *WOString* appears in the client browser just as it appears in the WebObjects application source code. If escape HTML is set to NO, WebObjects simply passes the string to the browser without protecting HTML tags from being interpreted as commands.

2 Click the *Delete* button to remove a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

WOStateStorage

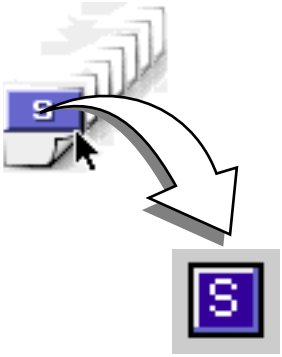
The *WOStateStorage* icon inserts an element that provides a simple mechanism for storing application states in an HTML page. If you include a *WOStateStorage* element in a form, any session and persistent data will be stored in the page rather than on the server.

Because *WOStateStorage* elements are implemented using hidden fields–which in HTML must be located within a form–they too must be located within a form. If a page has more than one form, you must declare a *WOStateStorage* element within each form.

To insert a *WOStateStorage* tag, proceed as follows:

1 Drag the *WOStateStorage* tag icon from the *Palette* and drop it in your layout grid or document window.

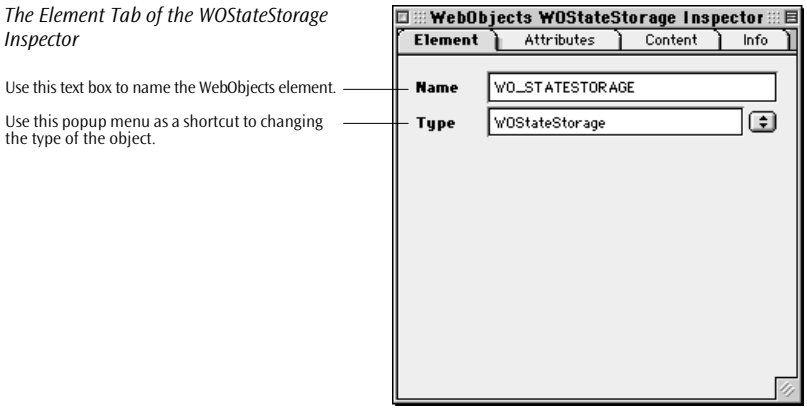
Inserting a WOStateStorage Tag



2 Select the *WOStateStorage* tag to set up the state storage element in the context-sensitive *Inspector* window, now titled *WOStateStorage Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

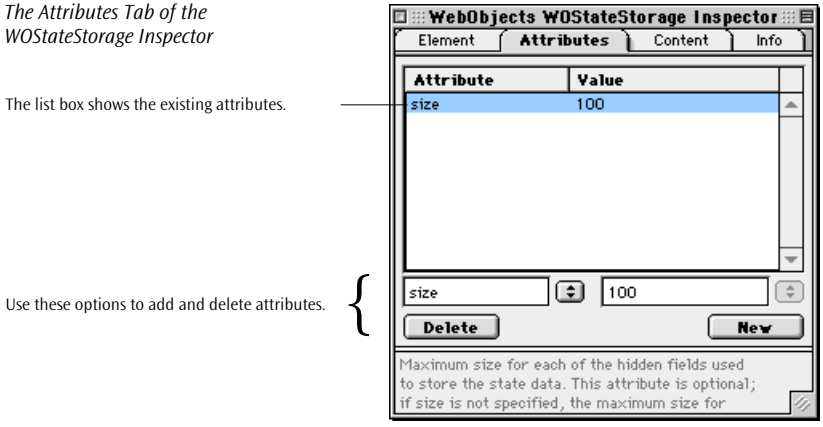
The Element Tab of the WOStateStorage Inspector



Set the following options in the *Element* tab of the *WOStateStorage Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOStateStorage* element as a unique entity in case more state storage elements follow in the HTML template page.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if required.

The Attributes Tab of the WOStateStorage Inspector



The *Attributes* tab of the *WOStateStorage Inspector* lets you add new attributes and values, as required.

To add and delete attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOStateStorage* object. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute

name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.

Currently supported attributes include:

- *size*: This attribute indicates the maximum size (in bytes) for each of the hidden fields used to store the state data. This attribute is optional; if size is not specified, the maximum size for hidden fields will be 1000 bytes.

2 Click the *Delete* button to delete a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

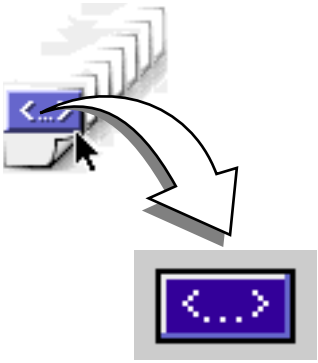
WOGenericElement

The *WOGenericElement* icon inserts an element that provides a way for WebObjects to accommodate custom HTML tags that are empty; in other words, they do not span a range of text. Because the HTML language is evolving rapidly, it is convenient to have a way to dynamically generate elements that WebObjects does not explicitly support.

To insert a *WOGenericElement* tag, proceed as follows:

1 Drag the *WOGenericElement* tag icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOGenericElement Tag



2 Select the *WOGenericElement* tag to set up the generic element in the context-sensitive *Inspector* window, now titled *WOGenericElement Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.

- *Info* briefly describes the WebObjects element.

The Element Tab of the WOGenericElement Inspector

Set the following options in the *Element* tab of the *WOGenericElement Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOGenericElement* element as a unique entity in case more unary unknown HTML elements follow in the page.
- Use the *Type* text box and popup menu to select a custom WebObjects type from the database. (See *Editing the WebObjects Inventory in the Web Database* on page 6.)

The Attributes Tab of the WOGenericElement Inspector

The *Attributes* tab of the *WOGenericElement Inspector* lets you add new attributes and values.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOGenericElement*. Clicking the *New* button enables the two text boxes and popup menus below the list box, allowing you to specify an attribute name (left) and an HTML tag as a value (right).
2. Select *elementName* (see below) from the left popup menu.

   Currently supported attributes include:

   - *elementName*: This is the name of the HTML element to generate. This name (for example, "HR") is used to generate the element's tag (<HR>). *elementName* must be defined as a constant. It cannot be something returned by a script method, for example. Please note that for elements with URL attributes, the URLs specified will appear as they are in the HTML document.
3. Go to the right popup menu and select an HTML tag—for example, "HR" for a horizontal ruler. Adobe GoLive's *Web Database* supplies the options on this popup menu.
4. To edit an attribute, click the *Delete* button to remove the selection from the *Attribute* list box, then start over.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

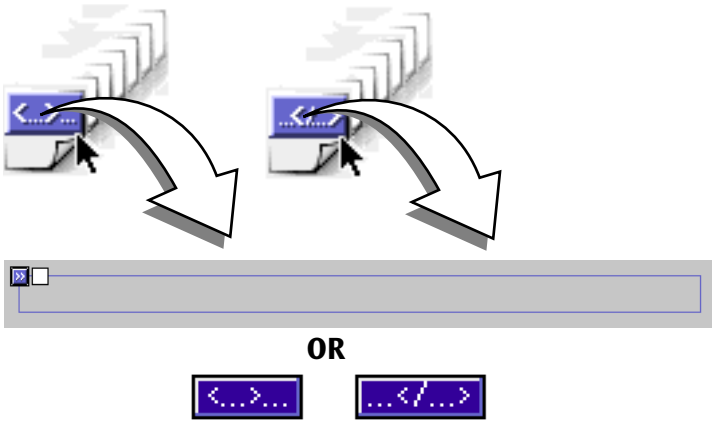
WOGenericContainer

The *WOGenericContainer* icon inserts a box-shaped container that provides a way for WebObjects to accommodate custom HTML container elements—that is, elements that span a range of text. Because the HTML language is evolving rapidly, it is convenient to have a way to dynamically generate elements that WebObjects does not explicitly support.

To insert a *WOGenericContainer* tag, proceed as follows:

1. Drag the *WOGenericContainer (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window.
2. Drag the item(s) you want to be enclosed into the box-shaped container.

Inserting a WOGenericContainer Tag

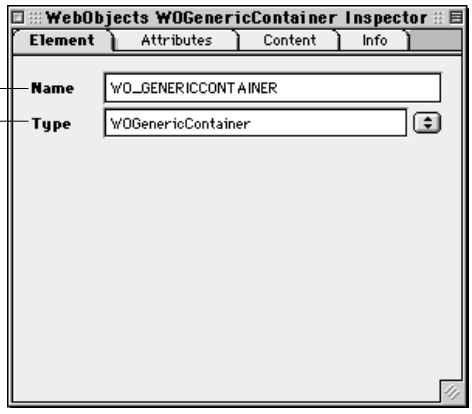


Tip: The *WOGenericContainer* element has two display options. You can change its appearance in the WebObjects Preferences (see page 107). If you do so, the default box-shaped container will be replaced by the two icons shown above, which must enclose the item(s) you want to wrap up in the *WOGenericContainer* element. An example appears on page 108.

3 Select the *WOGenericContainer* box or *WOGenericContainer (Begin)* tag to set up the generic container in the context-sensitive *Inspector* window, now titled *WOGenericContainer Inspector.* It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

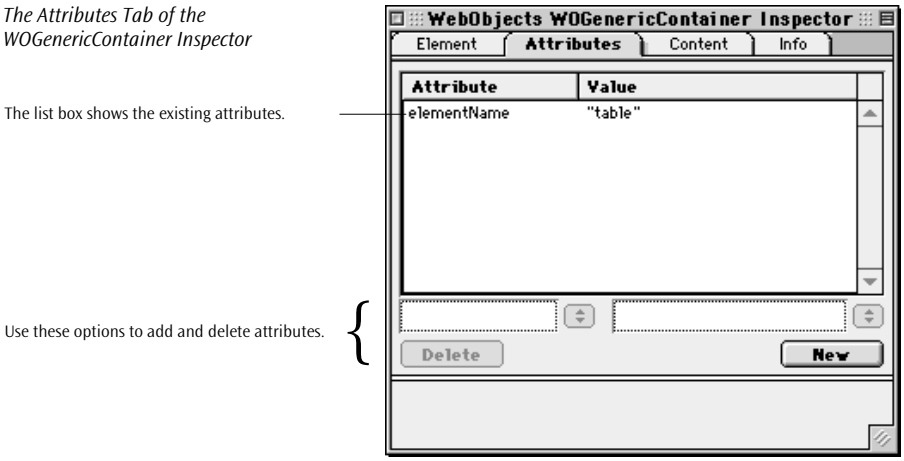
The Element Tab of the WOGenericContainer Inspector



Set the following options in the *Element* tab of the *WOGenericContainer Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOGenericContainer* object as a unique entity in case more unknown binary HTML elements follow in the HTML template page.

The Attributes Tab of the WOGenericContainer Inspector



The *Attributes* tab of the *WOGenericContainer Inspector* lets you add attributes and values, as necessary.

To add and delete attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOGenericContainer* element. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press the Return key to confirm your entry and add the new attribute to the list.

   Currently supported attributes include:

   - *elementName*: This attribute indicates the name of the HTML element to generate. This name (for example, “TABLE”) is used to generate the container’s opening and closing tags. *elementName* cannot be something returned by a script method, for example. Please note that for elements with URL attributes, the URLs specified will appear as they are in the HTML document.

Tip: Specifying the *elementName* attribute activates the popup menu above the *New* button. This menu offers a complete list of HTML tags, which can be used as attribute values.

2 Click the *Delete* button to delete a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

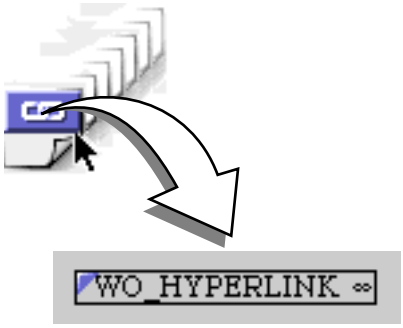
WOHyperlink

The *WOHyperlink* icon inserts an element that generates a hypertext link in an HTML document.

To insert a *WOHyperlink* tag, proceed as follows:

1 Drag the *WOHyperlink* tag icon from the *Palette* and drop it in your layout grid or document window.

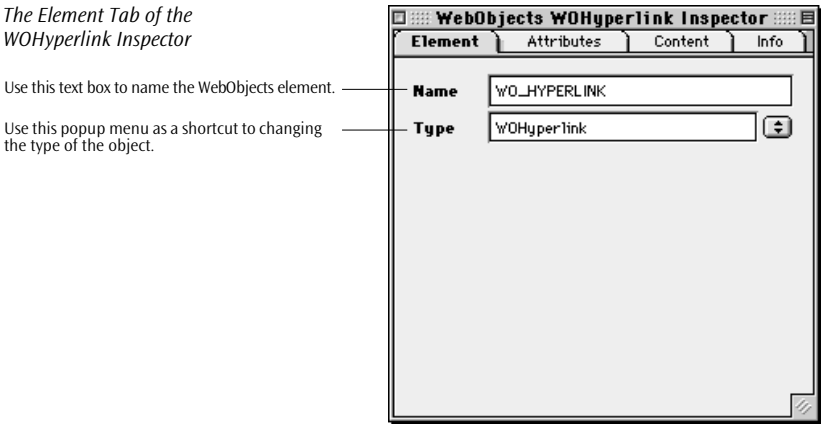
Inserting a WOHyperlink Tag



2 Select the *WOHyperlink* tag to set up the hyperlink element in the context-sensitive *Inspector* window, now titled *WOHyperlink Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

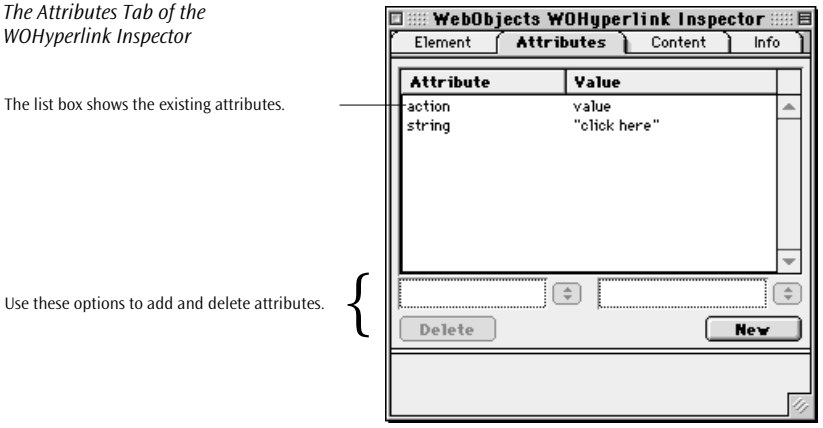
The Element Tab of the WOHyperlink Inspector



Set the following options in the *Element* tab of the *WOHyperlink Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOHyperlink* element as a unique entity in case more unary unknown HTML elements follow in the page.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

The Attributes Tab of the WOHyperlink Inspector



The *Attributes* tab of the *WOHyperlink Inspector* lets you add new attributes and values.

To add and delete attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOHyperlink* element. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left)

and value (right). Press Return to confirm your entry and add the new attribute to the list.

Currently supported attributes include:

- *action*: This attribute specifies the method to invoke when this element is activated. The method must return a WOElement.
- *href*: This attribute specifies the URL to direct the browser to when the image is clicked.
- *pageName*: This attribute specifies the *Name* of the WebObjects page to display when the link is clicked.
- *fragmentIdentifier*: This attribute specifies the named location to display in the destination page.
- *string*: This attribute specifies the text displayed to the site visitor as the link.
- *target*: This attribute specifies the frame in a frame set that will receive the page returned as a result of the site visitor's click.
- *disabled*: If this attribute is set to YES, the content string is displayed, but the hyperlink is not active.

2 Click the *Delete* button to delete a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

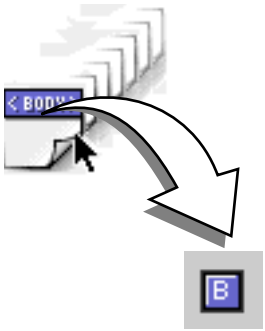
WOBody

The *WOBody* icon inserts an element that specifies the background image to display for the HTML page.

To insert a *WOBody* tag, proceed as follows:

1 Drag the *WOBody* tag icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOBody Tag

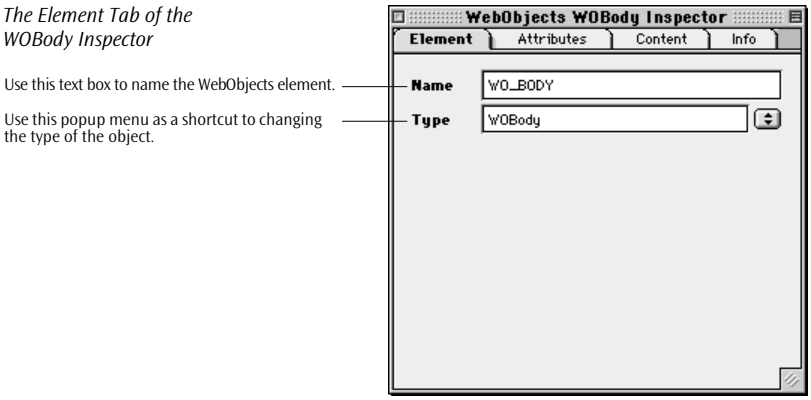


2 Select the *WOBody* tag to set up the background image in the context-sensitive *Inspector* window, now titled *WOBody Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays the image content.
- *Info* briefly describes the WebObjects element.

Tip: To avoid conflicts, disable the default HTML body tag in the page. In *Layout* mode, click the small page icon above the main window area to open the *Page Inspector*. Go to the *HTML* tab and deselect the *Body (if empty)* checkbox.

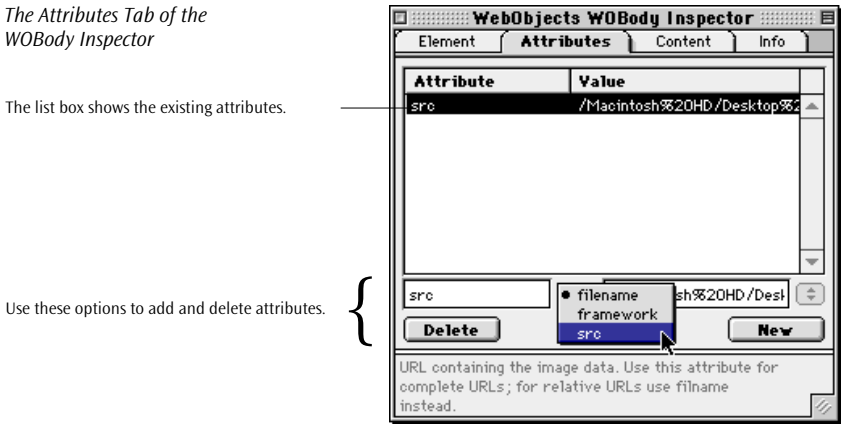
The Element Tab of the WOBody Inspector



Set the following options in the *Element* tab of the *WOBody Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

The Attributes Tab of the WOBody Inspector



The *Attributes* tab of the *WOBody Inspector* lets you add new attributes and values or change the image file reference.

To add and change attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOBody* element. Clicking the *New* button enables the two text boxes below the list box, allowing you to type the attribute name (left) or select it from the popup menu to the right of the text box. After specifying the attribute, enter its value (right), that is, the file specification. Press Return to confirm your entry and add the new attribute to the list.

   Currently supported attributes include:

   - *filename*: Path to the image relative to the WebServerResources directory.
   - *framework*: Framework that contains the image file. This attribute is only necessary if the image file is in a different location from the component.
   - *src*: URL containing the image data. Use this attribute for complete URLs; for relative URLs use filename instead.

2. Click the *Delete* button to remove a selection from the *Attribute* list box.

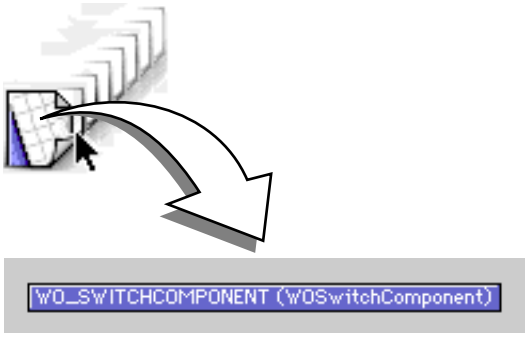
WOSwitchComponent

The *WOSwitchComponent* element provides a way to determine at runtime which nested component to display. This component is useful when you want to decide how to display information based on the state of the application.

To insert a *WOSwitchComponent* tag, proceed as follows:

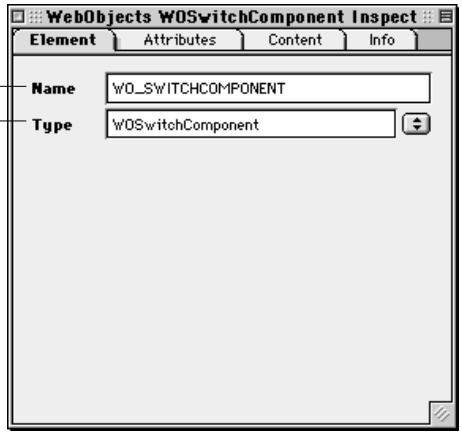
1. Drag the *WOSwitchComponent* tag icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOSwitchComponent Tag



2. Select the *WOSwitchComponent* tag to set up the background image in the context-sensitive *Inspector* window, now titled *WOSwitchComponent Inspector*. It has four tabs:
   - *Element* lets you set general properties for the current WebObjects.
   - *Attributes* lets you edit attributes for the current WebObjects.
   - *Content* displays the component content.
   - *Info* briefly describes the WebObjects element.

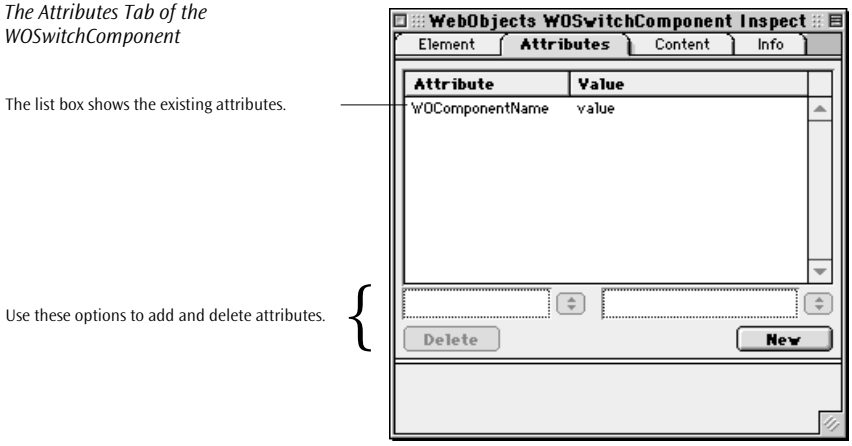
The Element Tab of the WOSwitchComponent Inspector



Set the following options in the *Element* tab of the *WOSwitchComponent Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

The Attributes Tab of the WOSwitchComponent



The *Attributes* tab of the *WOSwitchComponent Inspector* lets you add new attributes and values or change the image file reference.

To add and change attributes, proceed as follows:

1 Click the *New* button to add new attributes for the *WOBody* element. Clicking the *New* button enables the two text boxes below the list box, allowing you to type the attribute name (left) or select it from the popup menu to the right of the text box. After specifying the attribute, enter its value (right), that is, the file specification. Press Return to confirm your entry and add the new attribute to the list.

Currently, only one attribute is supported:

- *WOComponentName*: Name of the component to display. This attribute can be a string or a method that returns the name of a component.

2 Click the *Delete* button to delete a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

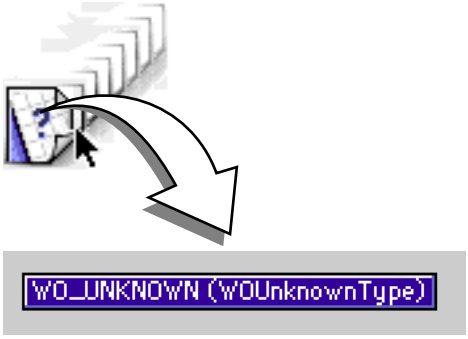
ReusableComponent

The *ReusableComponent* icon inserts an unknown dynamic element that can be linked with any of the reusable components that WebObjects 3.5 supplies. For a complete listing of reusable components that Adobe GoLive currently supports, please refer to the *WebObjects* tab of the *Web Database*.

To insert a *WOUnknownType* element, proceed as follows:

1 Drag the *ReusableComponent* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOUnknownType Element



2 Select the *WOUnknownType* tag to set up the unknown element in the context-sensitive *Inspector* window, now titled *WOUnknownType Inspector*. It has four tabs:

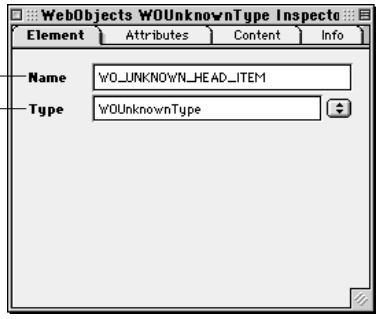
- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

3 Control-double-click the icon in the document window to open the reusable component for editing.

The Element Tab of the WOUnknownType Inspector

Use this text box to name the WebObjects element.

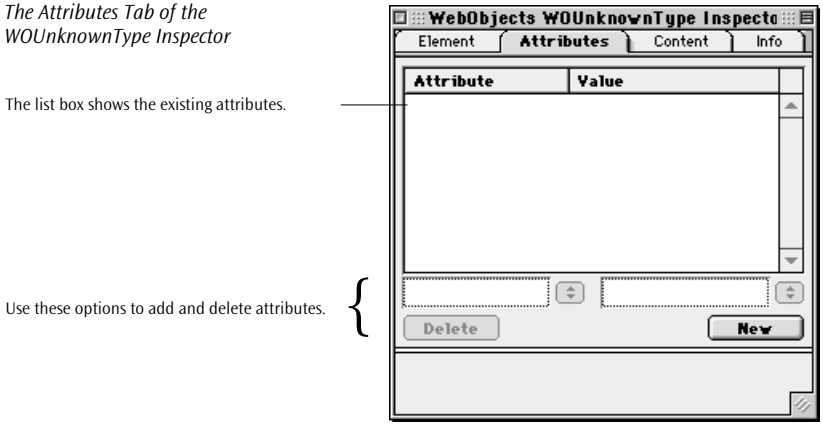
Use this popup menu as a shortcut to changing the type of the object.



Set the following options in the *Element t*ab of the *WOUnknownType Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This name identifies the *WOUnknownType* element as a unique entity in case more unknown WebObjects elements follow in the page.
- Use the *Type* text box and popup menu to select a custom WebObjects type from the database. (See *Editing the WebObjects Inventory in the Web Database* on page 6.)

The Attributes Tab of the WOUnknownType Inspector



The *Attributes* tab of the *WOUnknownType Inspector* lets you add new attributes and values.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOUnknownType* element. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attribute* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

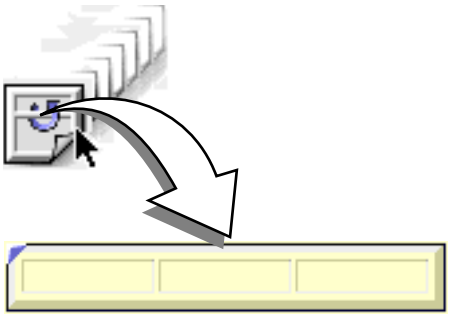
Table

The *Table with Repetition* icon inserts a table, providing a way for WebObjects to accommodate HTML tables in HTML template pages.

To insert a *Table with Repetition*, proceed as follows:

1 Drag the *Table with Repetition* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOTable Element



2 Select the *Table with Repetition* tag to set up the table in the context-sensitive *Inspector* window, now titled *Table Inspector*.

For a complete description of the *Table Inspector*, please refer to Chapter 4 of the *User Guide*. For an update on new shortcuts and features, please see the *Shortcuts* section at the end of the *User Guide*.

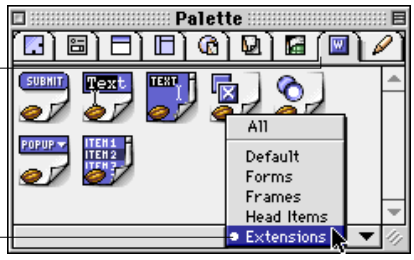
WebObjects Client-Side Components

The WebObjects extensions of Adobe GoLive each reference a special Java applet to include it in a WebObjects application. These applets are called "client-side components". The client-side components feature includes a suite of ready-to-use Java applets that display common controls (button, text field, etc.) on a Web page.

Displaying WebObjects Extensions

The *WebObjects Extensions* inventory resides in the *Extensions* view of the *WebObjects* tab. To display this view, select the *Extensions* item from the view control menu at the bottom of the *Palette*.

The WebObjects Tab of the Palette—Extensions View Selected



To view the WebObjects Extensions inventory, open the WebObjects tab of the Palette, …

… then select Extensions from the Palette's view control menu.

All of the elements in the *Extensions* view represent special cases of the basic *WOApplet* element described earlier (see page 24). However, while *WOApplet*s reside on the server, these Java applets are preloaded onto the client machine at runtime to allow communication with the server.

WOApplets and client-side controls share the same *Inspector* window and, hence, the same set of parameters, the only difference being that each of these elements is connected with a dedicated Java class.

Unlike the generic *WOApplet* element, the elements in the *Extensions* view all have two extra attributes, *archiveNames* and *associationClass*, which default to "WOExtensions" and "next.wo.client.SimpleAssociation.class", respectively. The *archiveNames* attribute specifies archive files that contain all the Java classes necessary to implement client-side components, while the *associationClass* attribute differentiates the client-side components from any other applet you might include in your application.

For more details on client-side components, please refer to the WebObjects documentation.

The following pages present the WebObjects extensions that the *Palette* offers, sorted in the order of their appearance:

- *WOButtonApplet* (see page 58)
- *WORadioGroupApplet* (see page 61)
- *WOCheckboxApplet* (see page 64)
- *WOChoiceApplet* (see page 66)
- *WOListApplet* (see page 69)
- *WOScrollingTextApplet* (see page 72)
- *WOTextFieldApplet* (see page 75)

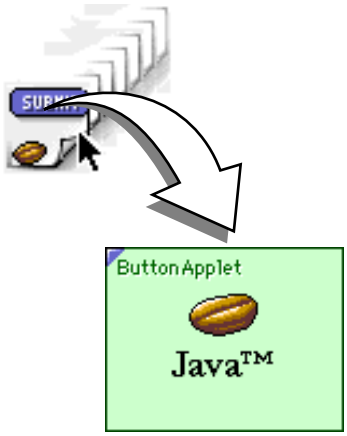
WOButtonApplet

The *WOButtonApplet* icon inserts a dynamic element that generates HTML to specify a Java-driven *Submit* button.

To insert a *WOButtonApplet* placeholder, proceed as follows:

1. Drag the *WOButtonApplet* icon from the *Palette* and drop it in your layout grid or document window.

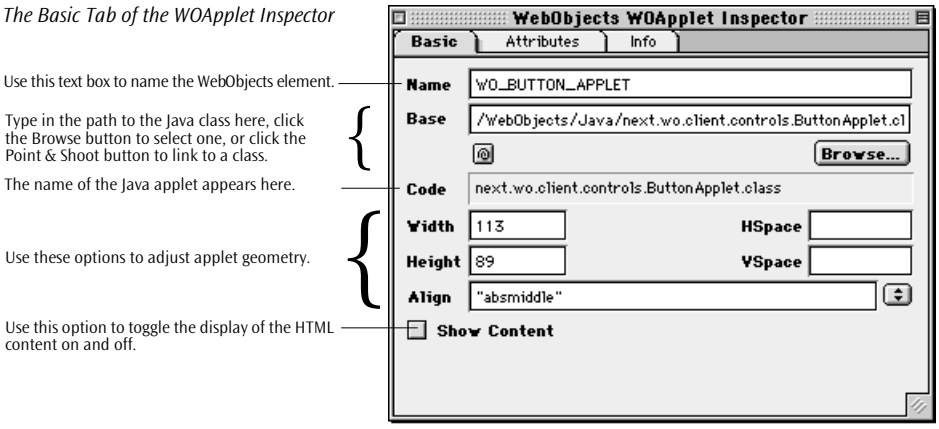
Inserting a WOButtonApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WOButtonApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



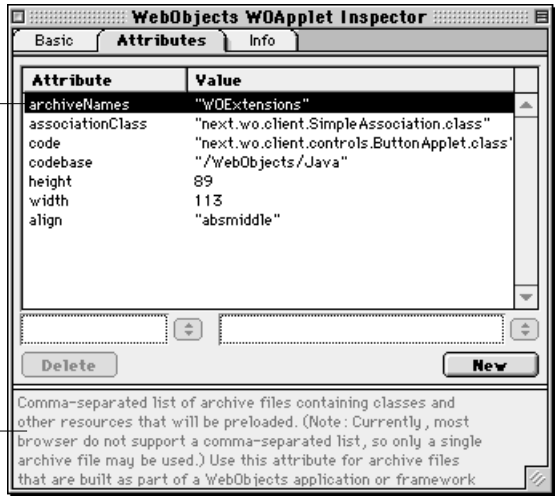
Set the following options in the *Basic* tab of the *WOApplet Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.
- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The list box shows the existing attributes.

Use these options to add and delete attributes.

Attribute descriptions appear in this text field.

The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings made in the *Basic* tab and add new attributes as required.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

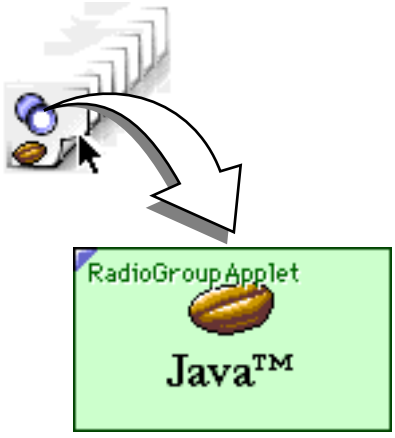
WORadioGroupApplet

The *WORadioGroupApplet* icon inserts a dynamic element that generates HTML to specify a Java-driven vertical matrix of checkboxes of which only one box can be checked at a time.

To insert a *WORadioGroupApplet* placeholder, proceed as follows:

1. Drag the *WORadioGroupApplet* icon from the *Palette* and drop it in your layout grid or document window.

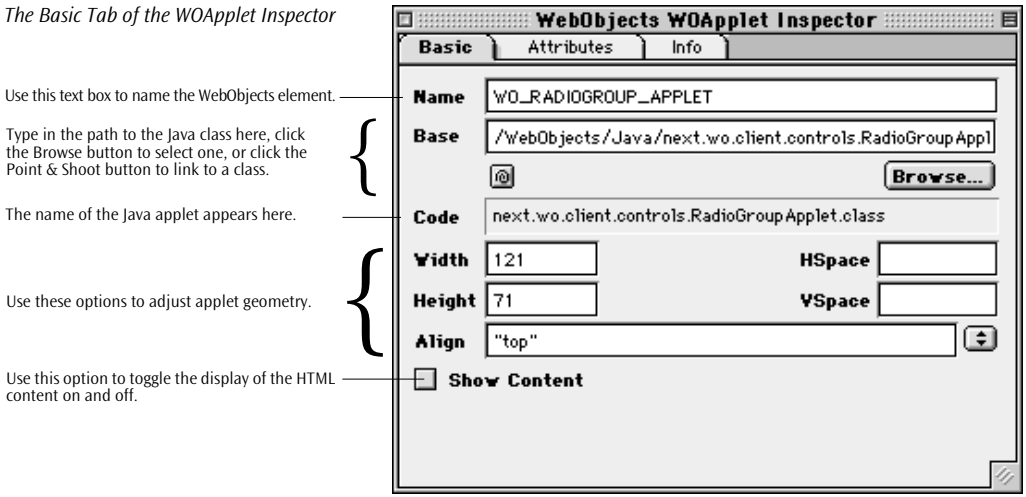
Inserting a WORadioGroupApplet



2. Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WORadioGroupApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



Set the following options in the *Basic* tab of the *WOApplet Inspector*:

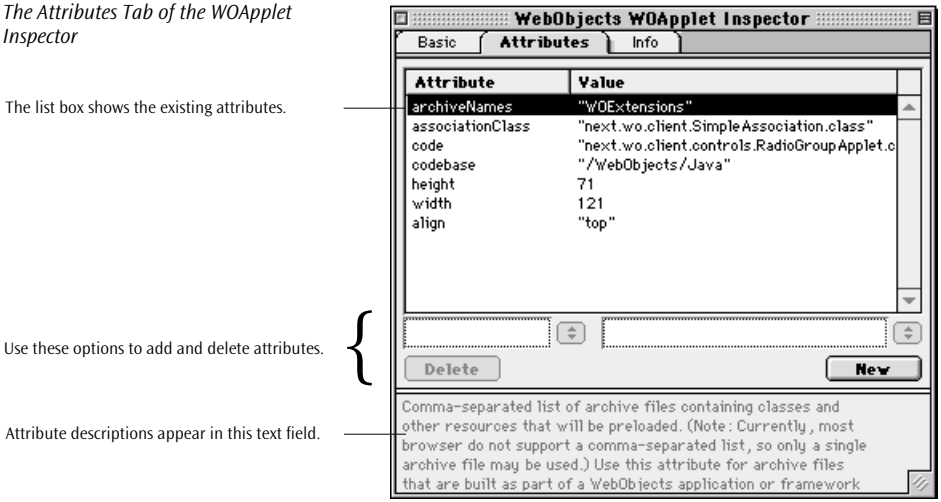
- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.

- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.
- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as necessary.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

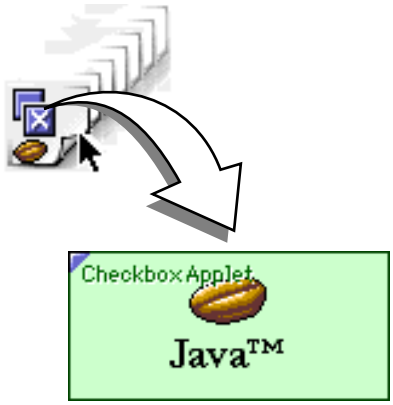
WOCheckboxApplet

The *WOCheckboxApplet* icon inserts a dynamic element that generates HTML to specify a Java-driven control that uses an image of a check box to indicate “off” and “on” states.

To insert a *WOCheckboxApplet* placeholder, proceed as follows:

1 Drag the *WOCheckboxApplet* icon from the *Palette* and drop it in your layout grid or document window.

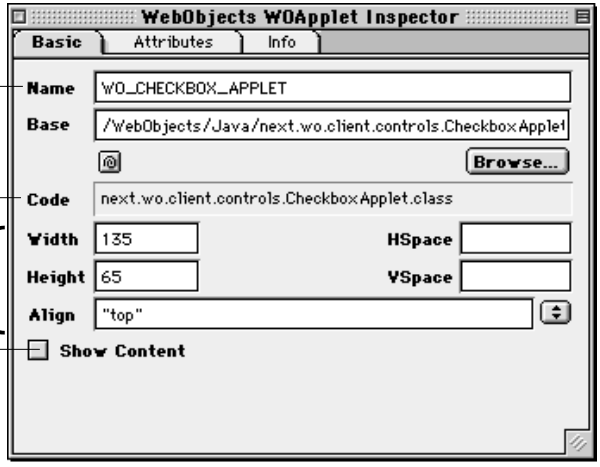
Inserting a WOCheckboxApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WOCheckboxApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



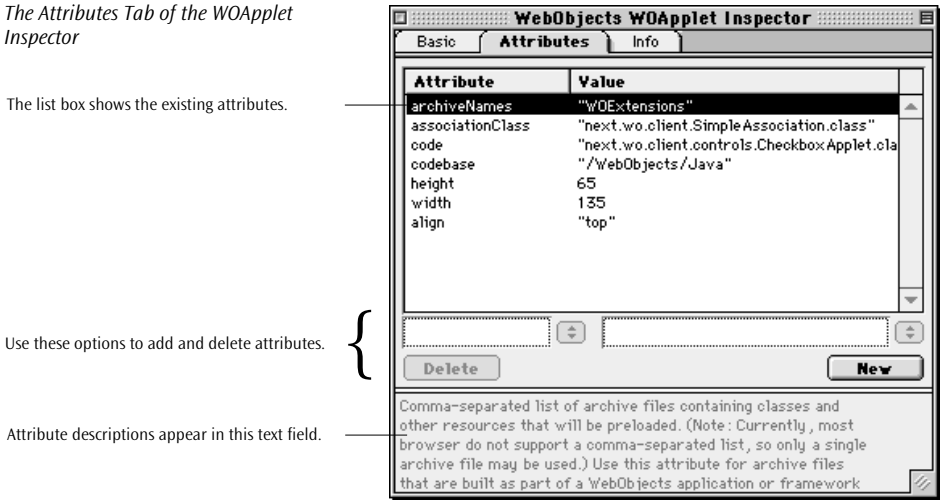
Set the following options in the *Basic* tab of the *WOApplet Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.
- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as needed.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to delete a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

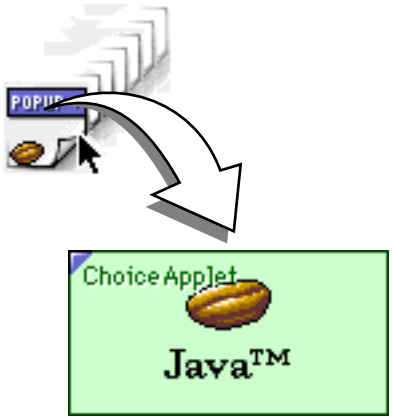
WOChoiceApplet

The *WOChoiceApplet* icon inserts a dynamic element that generates HTML to specify a non-editable list of items from which you can select one—for example, a popup menu.

To insert a *WOChoiceApplet* placeholder, proceed as follows:

1. Drag the *WOChoiceApplet* from the *Palette* and drop it in your layout grid or document window.

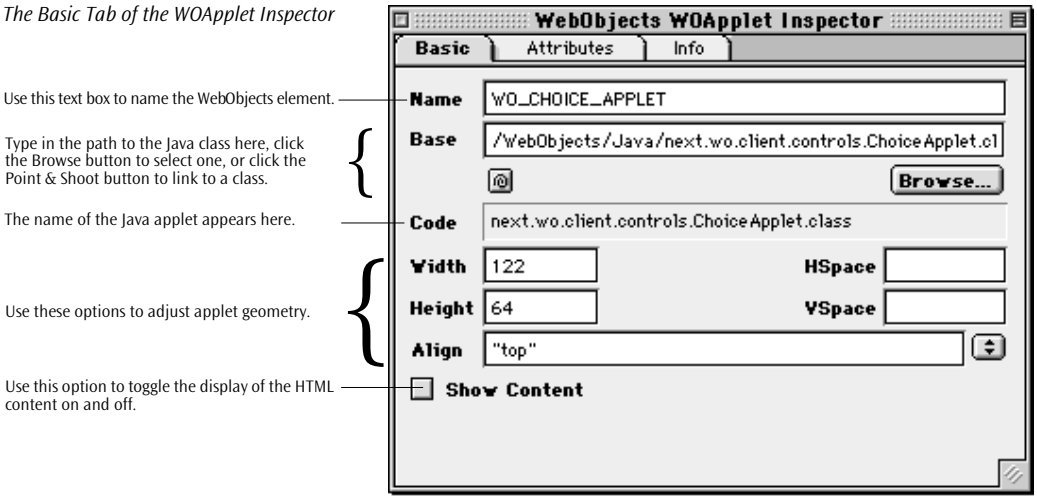
Inserting a WOChoiceApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WOChoiceApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



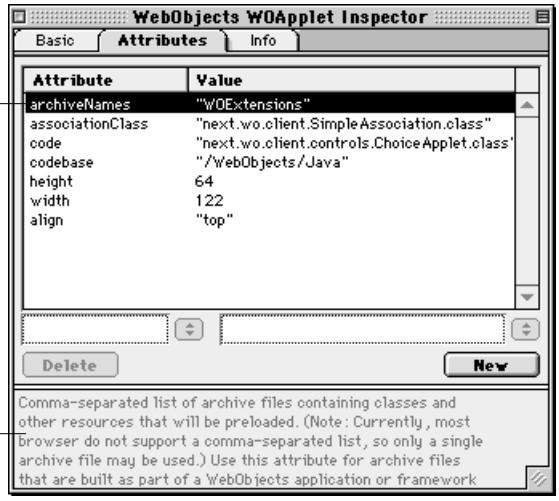
Set the following options in the *Basic* tab of the *WOApplet Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.
- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as necessary.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

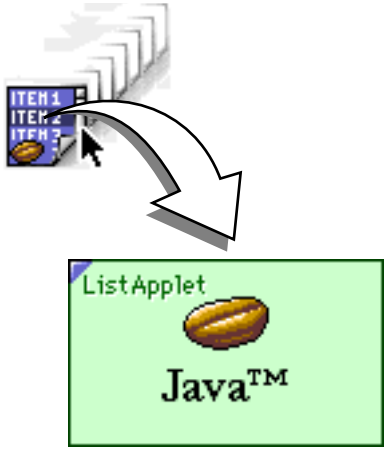
WOListApplet

The *WOListApplet* icon inserts a dynamic element that generates HTML to specify a non-editable list of items from which you can select one—for example, a popup menu.

To insert a *WOListApplet* placeholder, proceed as follows:

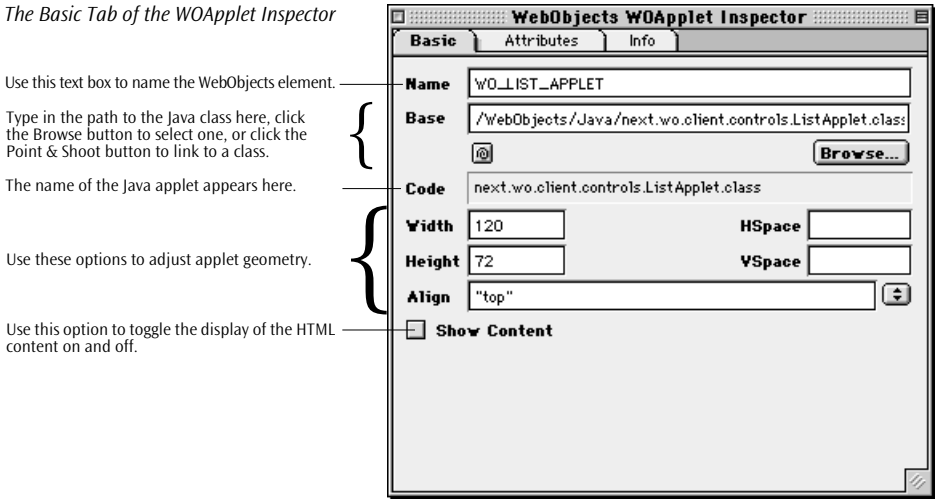
1. Drag the *WOListApplet* from the *Palette* and drop it in your layout grid or document window.

Inserting a WOListApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:
   - *Basic* lets you make basic settings for the *WOListApplet*.
   - *Attributes* allows you to inspect current attributes and add new ones.
   - *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



Set the following options in the *Basic* tab of the *WOApplet Inspector*:

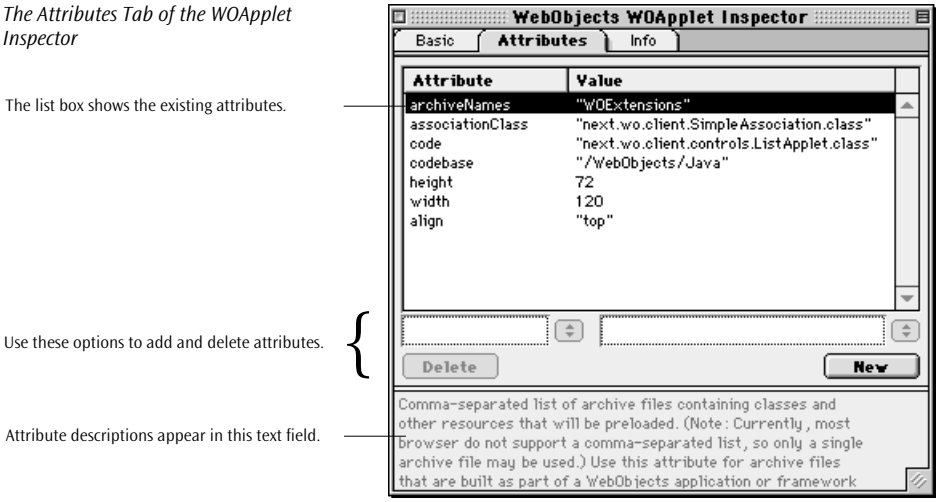
- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.

- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.
- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as needed.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

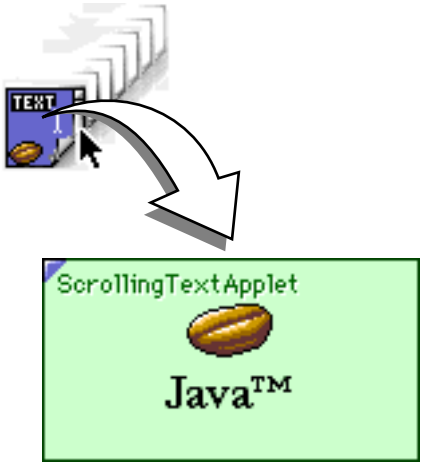
WOScrollingTextApplet

The *WOScrollingTextApplet* icon inserts a dynamic element that generates HTML to display a bordered area on the page. This area allows for viewing and editing one or more lines of text. If the amount of text exceeds the display area, scrollbars enable you to move text up and down within the display area.

To insert a *WOScrollingTextApplet* placeholder, proceed as follows:

1 Drag the *WOScrollingTextApplet* from the *Palette* and drop it in your layout grid or document window.

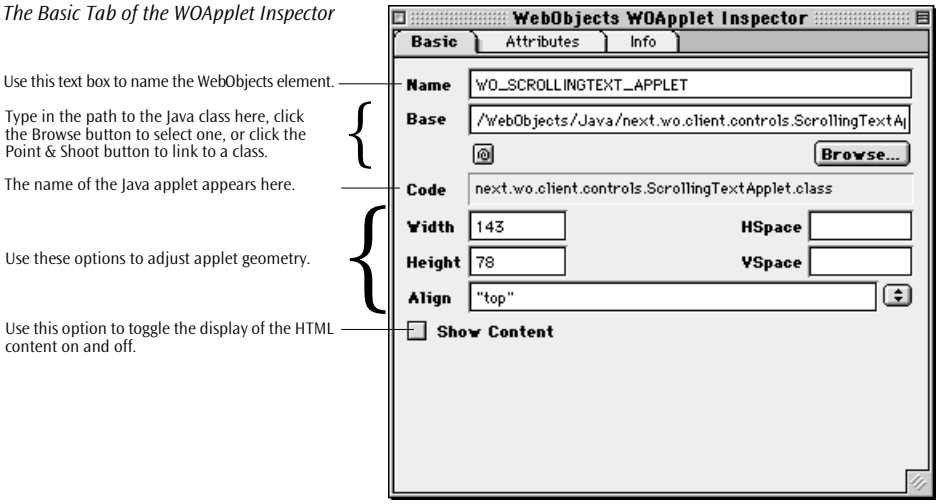
Inserting a WOScrollingTextApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WOScrollingTextApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



Set the following options in the *Basic* tab of the *WOApplet Inspector*:

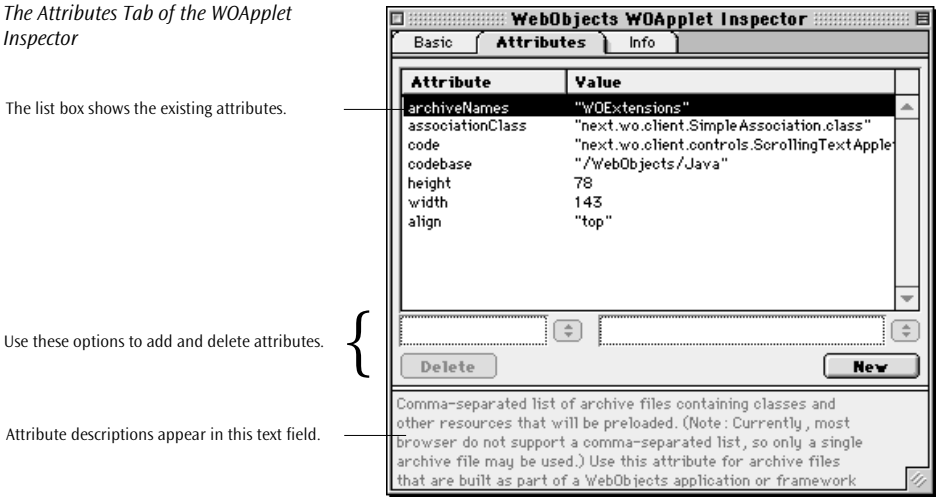
- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.

- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as necessary.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

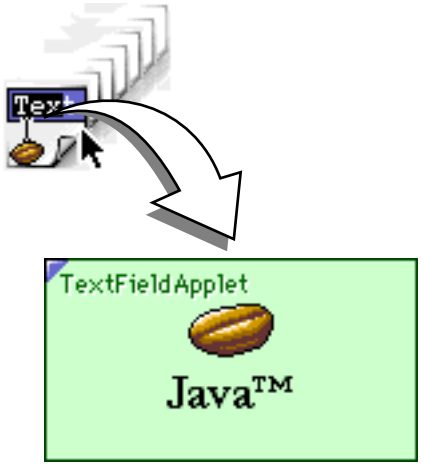
WOTextFieldApplet

The *WOTextFieldApplet* icon inserts a dynamic element that generates HTML to display a bordered area on the page. This area allows for viewing and editing one or more lines of text. If the amount of text exceeds the display area, scrollbars enable you to move text up and down within the display area.

To insert a *WOTextFieldApplet* placeholder, proceed as follows:

1 Drag the *WOTextFieldApplet* from the *Palette* and drop it in your layout grid or document window.

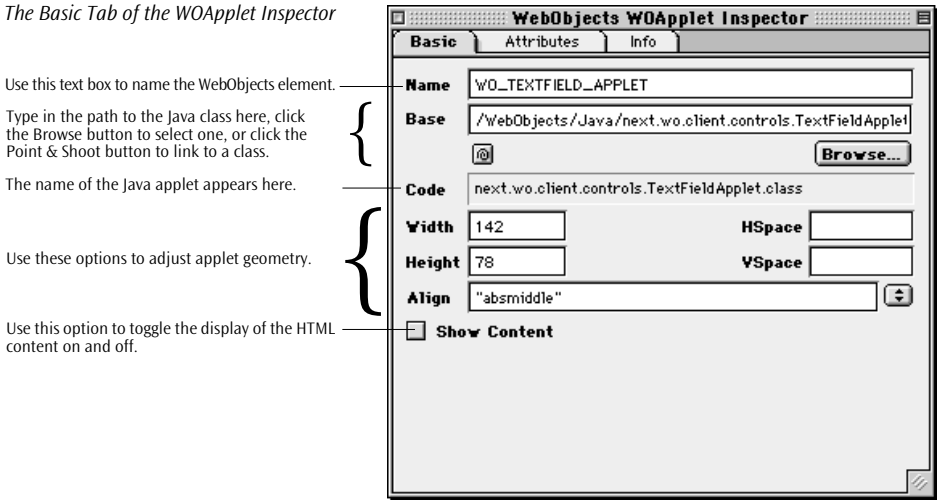
Inserting a WOTextFieldApplet Placeholder



2 Set up the dynamic element in the context-sensitive *Inspector* window, now titled *WOApplet Inspector*. It has three tabs:

- *Basic* lets you make basic settings for the *WOTextFieldApplet*.
- *Attributes* lets you inspect current attributes and add new ones.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOApplet Inspector



Set the following options in the *Basic* tab of the *WOApplet Inspector*:

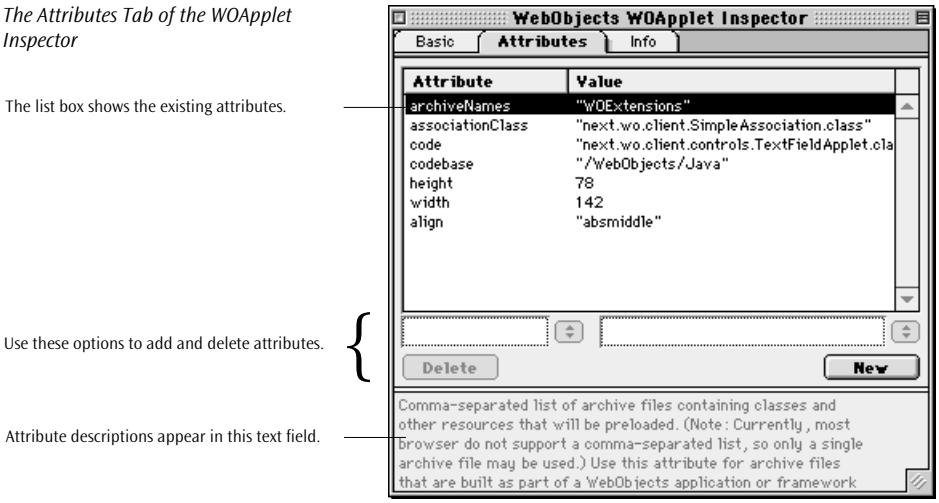
- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Base* text box to enter the path to the Java applet.

Alternatively, click the *Browse* button and select an applet in the following file selection dialog, or click the *Point & Shoot* button to link to an image in the *Site Window*.

- The *Code* text box displays the name of the Java class.
- Click to place the cursor in the *Width* text box, and enter the desired overall width of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Click to place the cursor in the *Height* text box, and enter the desired overall height of the area in pixels to allocate space for the applet. Press the Return key to confirm your entry.
- Use the *HSpace* text box to specify the horizontal spacing in pixels, and press the Return key to confirm your entry.
- Use the *VSpace* text box to specify the vertical spacing in pixels, and press the Return key to confirm your entry.
- Use the *Align* text box and popup menu to align the applet relative to text on the same line. For information on the options of the *Align* popup menu, please refer to the section *Adjusting WOImage Alignment* on page 14.

- Select the *Show Content* option to toggle the display of the HTML content on and off.

The Attributes Tab of the WOApplet Inspector



The *Attributes* tab of the *WOApplet Inspector* lets you inspect the settings you made in the *Basic* tab and add new attributes as necessary.

To add and delete attributes, proceed as follows:

1. Click the *New* button to add new attributes for the *WOApplet*. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and add the new attribute to the list.
2. Click the *Delete* button to remove a selection from the *Attributes* list box.

To view an attribute description, proceed as follows:

Select the desired attribute from the list. The description (if any) appears in the text field at the bottom of the *Inspector* window.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

WebObjects Forms

The *WebObjects* view of the *Forms* tab contains a choice of WebObjects forms elements. Like their HTML counterparts, WebObjects forms accept user input. However, WebObjects forms elements let designers and developers work together to create interactive fill-in forms that are assembled dynamically at runtime, depending on selections that the visitor makes.

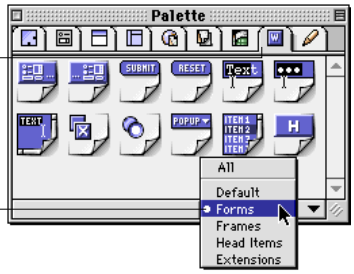
Displaying WebObjects Forms Tags

The WebObjects forms tag inventory resides in the *WebObjects* view of the *Forms* tab. This view can be displayed by selecting the *WebObjects* item from the view control menu at the bottom of the *Palette*.

The WebObjects Tab of the Palette—Forms View Selected

To view the WebObjects forms inventory, open the WebObjects tab of the Palette, …

… then select Forms from the Palette's view control menu.



The following pages describe the WebObjects forms elements that the *Palette* offers, sorted in the order of their appearance:

- *The WOForm Tags* (see page 78)
- *WOSubmitButton* (see page 81)
- *WOResetButton* (see page 82)
- *WOTextField* (see page 84)
- *WOPasswordField* (see page 86)
- *WOText* (see page 89)
- *WOHiddenField* (see page 91)
- *WOCheckbox* (see page 93)
- *WORadioButton* (see page 94)
- *WOPopupButton* (see page 96)
- *WOBrowser* (see page 98)

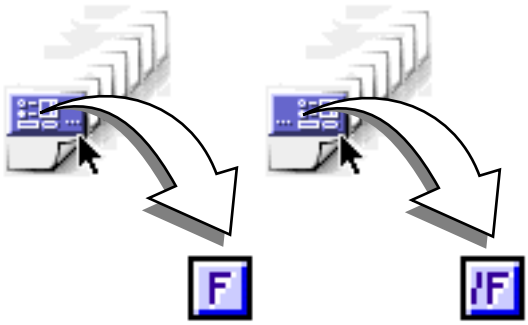
The WOForm Tags

Like its HTML equivalent, a WOForm is a container element that generates a fill-in form. The *WOForm* tag identifies the current page or section as a form and instructs the browser where and how to return form information for processing at runtime.

To insert a *WOForm* tag, proceed as follows:

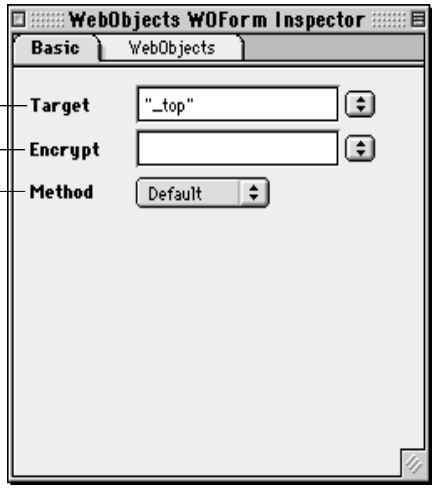
1. Drag the *WOForm (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window to mark the beginning of the form.
2. Drag the *WOForm (End)* tag icon from the *Palette* and drop it in your layout grid or document window to mark the end of the form.

Inserting a WOForm Tag



3 Select the *WOForm (Begin)* tag to set up the form in the context-sensitive *Inspector* window, now titled *WOForm Inspector*.

The Basic Tab of the WOForm Inspector



Select a target frame for the form output.

Use default encryption or select an encryption method from the popup menu.

Select a method for the form to return its information to the CGI script.

Set the following options in the *Basic* tab of the *WOForm Inspector*:

- Use the *Target* text box and popup menu to specify the frame in a frame set that will receive the page returned as a result of the site visitor's click.
- Use the *Encrypt* popup menu to select an encryption method.
- Use the *Method* popup menu to determine how to send the form information:
  - *Post* sends the form information separately from the destination URL.
  - *Get* appends the form information to the destination URL.
  - *Default* omits the *Method* attribute.

We recommend using the *Post* option because URLs have definite lengths that might be exceeded by simply appending information to the destination file, resulting in accidental loss of data.

The WebObjects Tab of the WOForm Inspector

Set the following options in the *WebObjects* tab of the *WOForm Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *Name* text box to name your WebObjects form element.
- Use the *Action* text box to specify an action method that is invoked when the form is submitted. If the form contains a dynamic element with its own action (such as a WOSubmitButton or a WOActiveImage), that action is invoked instead of the WOForm's action.
- Use the *HRef* text box to type in a URL specifying where the form will be submitted.

Alternatively, click the *Browse* button and select a destination in the following file selection dialog, or click the *Point & Shoot* button to link to a destination URL in the *Site Window*.

- Use the *Mult. Submit* text box and popup menu to determine whether the form can have more than one *WOSubmitButton*, each with its own action. By default, WOForm supports only a single *WOSubmitButton*. Setting *Mult. Submit* to *YES* enables multiple submit buttons.

Note: Some older browsers support only a single submit button in a form.

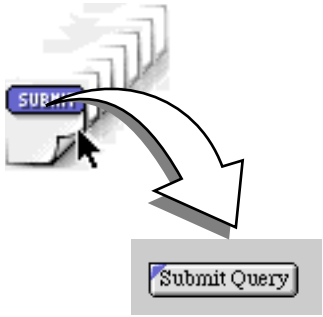
WOSubmitButton

The *WOSubmitButton* icon generates a submit button in an HTML form.

To insert a *WOSubmitButton*, proceed as follows:

1. Drag the *WOSubmitButton* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOSubmit Button



2. Set up the *Submit Button* in the context-sensitive *Inspector* window, now titled *WOSubmitButton Inspector*.

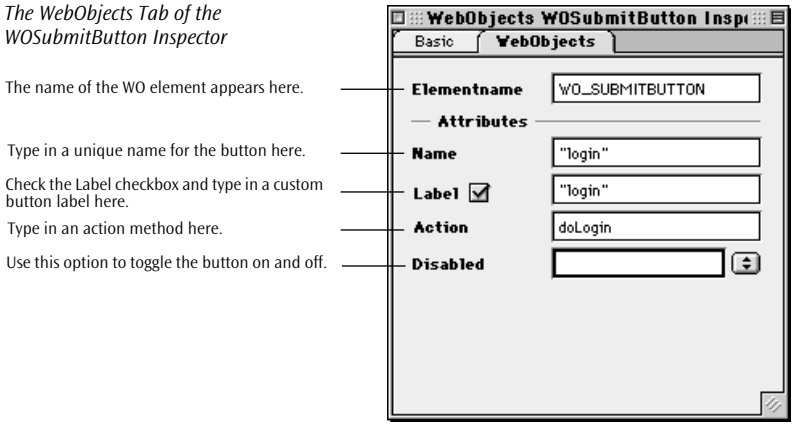
The Basic Tab of the WOSubmitButton Inspector

Click the appropriate radio button to toggle the button function.

Set the following options in the *WOSubmitButton Inspector*:

- Toggle between the *Submit* and *Reset* button functions by using the radio buttons.

The WebObjects Tab of the WOSubmitButton Inspector



Set the following options in the *WebObjects* tab of the *WOSubmitButton Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *Name* text box to give your WebObjects submit button a name that uniquely identifies this element within the form. You may specify a name or let WebObjects automatically assign one at runtime.
- Check the *Label* checkbox and type in the title of the button you want the audience to see—for example, *Login* instead of *Submit*.
- Use the *Action* text box to specify an action method that is invoked when the form is submitted.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.

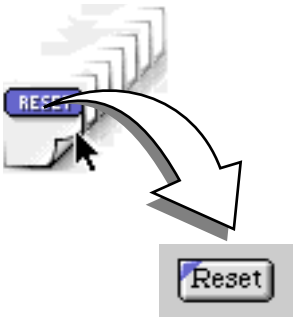
WOResetButton

The *WOResetButton* icon generates a reset button in an HTML form.

To insert a *WOResetButton*, proceed as follows:

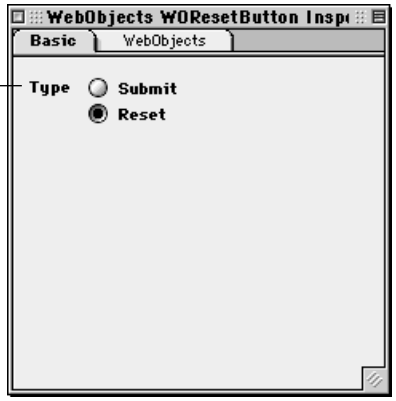
1 Drag the *WOResetButton* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOReset Button



2 Set up the *Reset Button* in the context-sensitive *Inspector* window, now titled *WOResetButton Inspector*.

The Basic Tab of the WOResetButton Inspector



Click the appropriate radio button to toggle the button function.

Set the following options in the *WOResetButton Inspector*:

- You can use the radio buttons to toggle between the *Submit* and *Reset* button functions.

The WebObjects Tab of the WOResetButton Inspector

The name of the WO element appears here.

Check the Label checkbox and type in a custom button label here.

Type in an action method here.

Use this option to toggle the button on and off.

Set the following options in the *WebObjects* tab of the *WOResetButton Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- The *Name* text box is disabled. A form can only have one reset button at a time.
- Check the *Label* checkbox and type in the title of the button you want the audience to see—for example, *Clear* instead of *Reset*.
- Use the *Action* text box to specify an action method that is invoked when the form is reset.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.

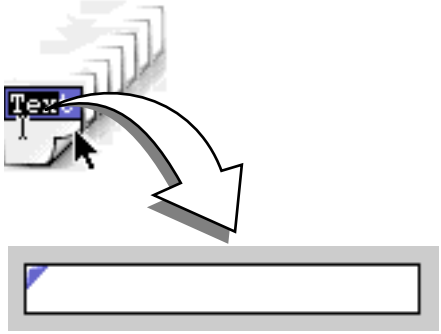
WOTextField

The *WOTextField* icon inserts a single-line text field that lets site visitors enter text—for example, their name or other personal data.

To insert a text field, proceed as follows:

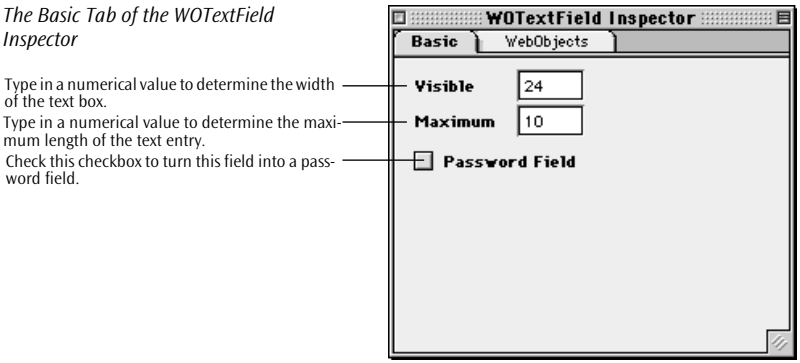
1. Drag the *WOTextField* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOTextField



2. Set up the text field in the context-sensitive *Inspector* window, now titled *WOTextField Inspector*.

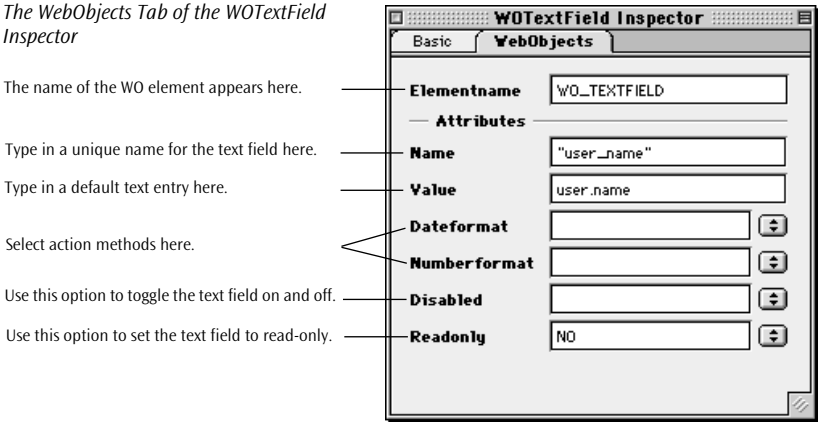
The Basic Tab of the WOTextField Inspector



Set the following options in the *Basic* tab of the *WOTextField Inspector*:

- In the *Visible* text box, type in the number of visible characters to determine the width of the text box.
- In the *Maximum* text box, type in the maximum number of characters that the text box accepts before truncation occurs. If empty, this limit is determined by the Web browser used to view the form.
- Check the *Password Field* checkbox to convert the text field into a password field, if necessary.

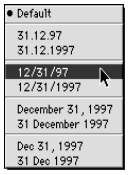
The WebObjects Tab of the WOTextField Inspector



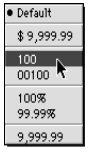
Set the following options in the *WebObjects* tab of the *WOTextField Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.

- Use the *Name* text box to give your WebObjects text field a name that uniquely identifies this element within the form. You may specify a name or let WebObjects automatically assign one at runtime.
- Use the *Value* text box to enter a default value displayed in the single-line text field while the page is being built. During request handling, this field holds the value the site visitor entered or the default value if the site visitor left the field untouched.



- Use the *Dateformat* text box and popup menu to select a format string that specifies how to format *Value* as a date. If you use a date format, *Value* must be assigned an NSCalendarDate object. If *Value* can't be interpreted according to the format you specify, *Value* is set to nil. See the WebObjects documentation for a description of the date format syntax.



- Use the *Numberformat* text box and popup menu to select a format string that specifies how to format *Value* as a number. If you use a number format, *Value* must be assigned an NSDecimalNumber object. If the element's *Value* can't be interpreted according to the format you specify, *Value* is set to nil. See the WebObjects documentation for a description of the number format syntax.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.
- Use the *Read-only* text box and popup menu to temporarily set the element to read-only. If set to YES, the element appears in the page but cannot be edited. A scripted method can override this property.

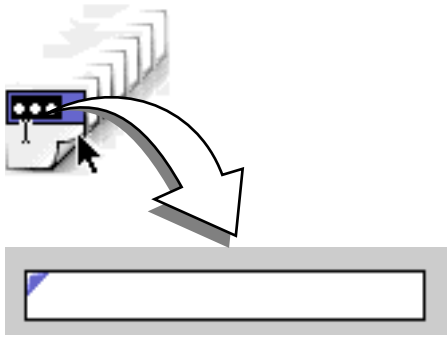
WOPasswordField

The *WOPasswordField* icon inserts a password field that lets the site visitor enter a password without echoing the characters typed.

To insert a password entry field, proceed as follows:

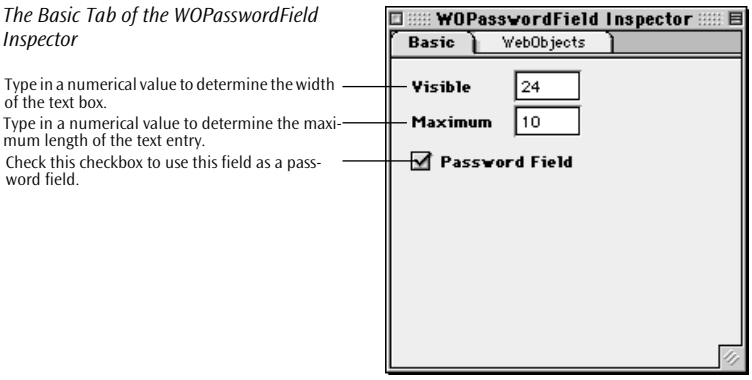
1 Drag the *WOPasswordField* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOPasswordField



2 Set up the password entry field in the context-sensitive *Inspector* window, now titled *WOPasswordField Inspector*.

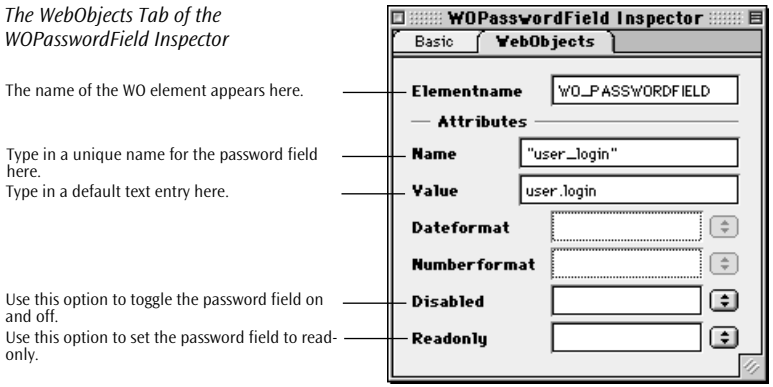
The Basic Tab of the WOPasswordField Inspector



Set the following options in the *Basic* tab of the *WOPasswordField Inspector*:

- In the *Visible* text box, type in the number of visible bullets (the password itself is not echoed) to determine the width of the text box.
- In the *Maximum* text box, type in the maximum number of characters that the text box accepts before truncation occurs. If empty, this limit is determined by the Web browser used to view the form.
- Make sure that the *Password Field* checkbox is checked.

The WebObjects Tab of the WOPasswordField Inspector



Set the following options in the *WebObjects* tab of the *WOPassword-Field Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *Name* text box to give your WebObjects password entry field a name that uniquely identifies this element within the form. You can specify a name or let WebObjects automatically assign one at runtime.
- Use the *Value* text box to enter a default value for the password field used while the page is being built. This value is not displayed to the site visitor. During request handling, this field holds the value the site visitor entered or the default value if the site visitor left the field untouched.
- The *Dateformat* text box and popup menu are disabled. Passwords are unformatted.
- The *Numberformat* text box and popup menu are disabled. Passwords are unformatted.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. You can override this status by a scripted method.
- Use the *Read-only* text box and popup menu to temporarily set the element to read-only. If set to YES, the element appears in the page but cannot be edited. A scripted method can override this property.

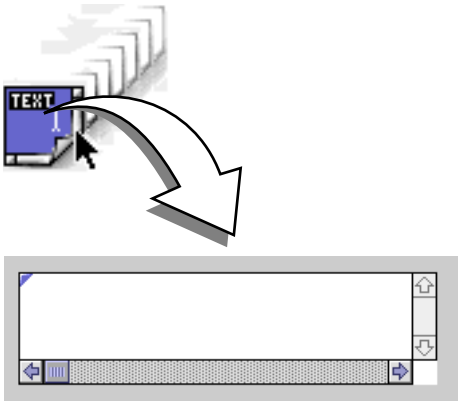
WOText

The *WOText* icon inserts a scrolling text area that lets the site visitor enter multiple lines of text—for example, feedback on your Web page.

To insert a WOText, proceed as follows:

1. Drag the *WOText* icon from the *Palette* and drop it in your layout grid or document window.

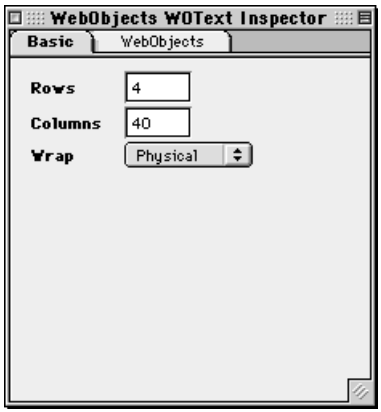
Inserting a WOText



2. Set up the *WOText* element in the context-sensitive *Inspector* window, now titled *WOText Inspector*.

The WOText Inspector

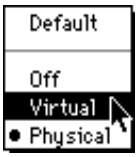
Use these options to resize the WOText element and control the behavior of the text.



Set the following options in the *Basic* tab of the *WOText Inspector*:

- In the *Rows* text box, type in the maximum number of rows to determine the height of the text box.
- In the *Columns* text box, type in the number of visible characters to determine the width of the *WOText* element.

- Choose the appropriate option from the *Wrap* popup menu to control the behavior of line breaks.
  - *Default* lets the browser use default settings for *WOText* objects.
  - *Off* instructs the browser to ignore the *Columns* limit and prevents text entered into the WOText from wrapping at the right margin of the box.
  - *Virtual* and *Physical* instruct the browser to respect the *Columns* limit. The entered text wraps when reaching the right margin of the box and starts scrolling vertically.



The WebObjects Tab of the WOText Inspector

Set the following options in the *WebObjects* tab of the *WOText Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *Name* text box to give your WebObjects password entry field a name that uniquely identifies this element within the form. You can specify a name or let WebObjects automatically assign one at runtime.
- The *Value* text box specifies the text or object that is displayed in the text field while the page is being built. During request handling, this field holds the text as the site visitor left it.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. You can override this status by a scripted method.
- Use the *Read-only* text box and popup menu to temporarily set the element to read-only. If set to YES, the element appears in the page but cannot be edited. A scripted method can override this property.

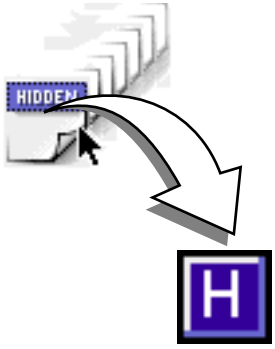
WOHiddenField

The *WOHiddenField* icon inserts a hidden field. Hidden fields are sometimes used to store application state data in an HTML page. In WebObjects, the *WOStateStorage* (see page 41) element is designed expressly for this purpose.

To insert a hidden field, proceed as follows:

1 Drag the *WOHiddenField* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOHiddenField Tag



2 Set up the *WOHiddenField* tag in the context-sensitive *Inspector* window, now titled *WOHiddenField Inspector*. It has four tabs:

- *Element* lets you set general properties for the current WebObjects.
- *Attributes* lets you edit attributes for the current WebObjects.
- *Content* displays object-specific text content, if any.
- *Info* briefly describes the WebObjects element.

The Element Tab of the WOHiddenField Inspector

Type in a unique name for the hidden tag here.

Use this popup menu as a shortcut to changing the type of the object.

Set the following options in the *Element* tab of the *WOHiddenField Inspector*:

- In the *Name* text box, type in a unique name to identify the hidden tag as an entity within the form. You can specify a name or let WebObjects automatically assign one at runtime.
- Use the *Type* text box and popup menu to change the type of the WebObjects, if necessary.

Caution: Use the *Type* option with care. You may lose your settings after you change the type of an object!

The Attributes Tab of the WOHiddenField Inspector

The list box shows the existing attributes.

Use these options to add and delete attributes.

Set the following options in the *Attributes* tab of the *WOHiddenField Inspector*:

- Click the *New* button to add new attributes for the hidden text field. Clicking the *New* button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press return to confirm your entry and add the new attribute to the list.
- Click the *Delete* button to delete a selection from the *Attributes* list box.

Note: Consult the latest release of the WebObjects documentation for a list of valid attributes and attribute values.

WOCheckbox

The *WOCheckbox* icon inserts a checkbox that lets the site visitor select multiple items from a list or make other selections to control processes.

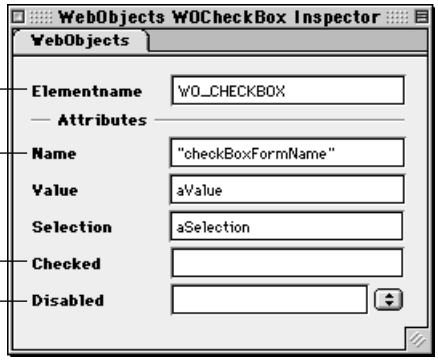
To insert a checkbox, proceed as follows:

1. Drag the *WOCheckbox* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOCheckbox

2. Set up the checkbox in the context-sensitive *Inspector* window, now titled *WOCheckbox Inspector*.

The WOCheckbox Inspector



Set the following options in the *WOCheckbox Inspector*:

- In the *Name* text box, type in a unique name to identify the checkbox as an entity within the form. You may specify a name or let WebObjects automatically assign one at runtime.
- In the *Value* text box, type in a value for this input element—for example, a variable. If not specified, WebObjects provides a default value.
- Use *Selection* to type in a string (for example, a variable) that causes WebObjects to check the checkbox automatically. If *Selection* and *Value* are equal when the page is generated, the check-

box is checked. When the page is submitted, *Selection* is assigned the value of the checkbox.

- Use the *Checked* text box and popup menu to select a default state for the checkbox. Setting *Checked* to YES causes the check box to appear in the checked state when the page is being generated. During request handling, *Checked* reflects the state the site visitor left the checkbox in: YES if checked; NO if not.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.

WORadioButton

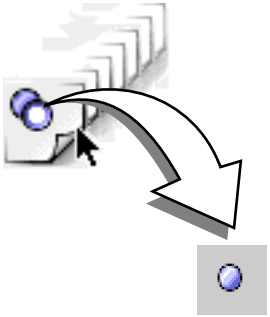
The *WORadioButton* icon inserts a radio button that acts as an on-off switch. Radio buttons are normally grouped because their main purpose is to allow the site visitor to chose exactly one of several options. If the site visitor selects one button, the previously selected button is deselected.

Because radio buttons usually appear as a group, *WORadioButton* is commonly placed within a *WORepetition* (see page 33).

To insert a radio button, proceed as follows:

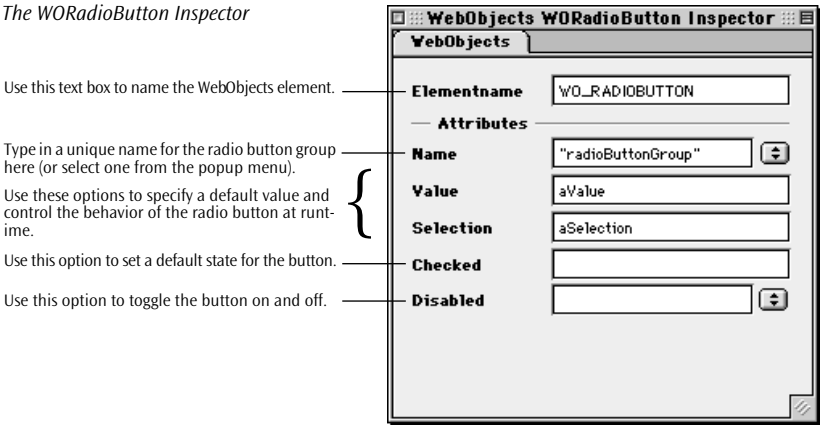
1 Drag the *WORadioButton* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WORadioButton



2 Set up the radio button in the context-sensitive *Inspector* window, now titled *WORadioButton Inspector*.

The WORadioButton Inspector



Set the following options in the *WORadioButton Inspector*:

- In the *Name* text box, type in a unique name that identifies the radio button's group. You can select only one radio button at a time within a group.
- In the *Value* text box, type in a value for this input element—for example, a variable. If not specified, WebObjects will provide a default value.
- Use *Selection* to type in a string (for example, a variable) that causes WebObjects to chose the radio button automatically. If *Selection* and *Value* are equal when the page is generated, the radio button is selected. When the page is submitted, *Selection* is assigned the value of the radio button.
- Use the *Checked* text box and popup menu to select a default state for the radio button. Setting *Checked* to YES causes the radio button to appear in the selected state when the page is being generated. During request handling, *Checked* reflects the state the site visitor left the radio button in: YES if selected; NO if not.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.

Note that a value for either *Checked* or *Value* is required in a *WORadioButton* declaration, but that they are mutually exclusive.

WOPopupButton

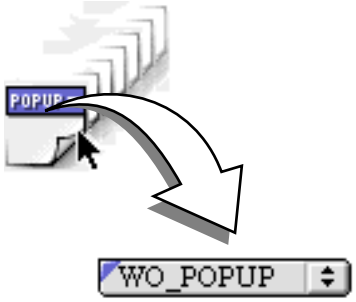
The *WOPopupButton* icon inserts a popup menu with multiple options to choose from. It displays itself as a selection list that allows the site visitor to pick only one item at a time.

The related element *WOBrowser* is similar to *WOPopUpButton* except that it allows the site visitor to select more than one item at a time.

To insert a *WOPopupButton* menu, proceed as follows:

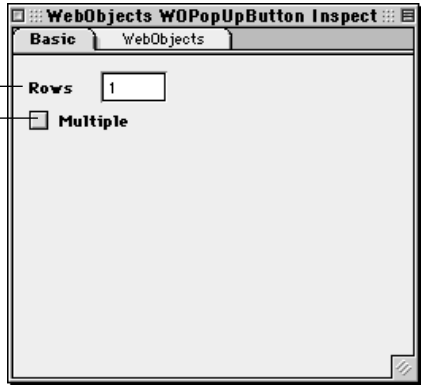
1. Drag the *WOPopupButton* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOPopupButton



2. Set up the list box in the context-sensitive *Inspector* window, now titled *WOPopupButton Inspector*.

The Basic Tab of the WOPopupButton Inspector



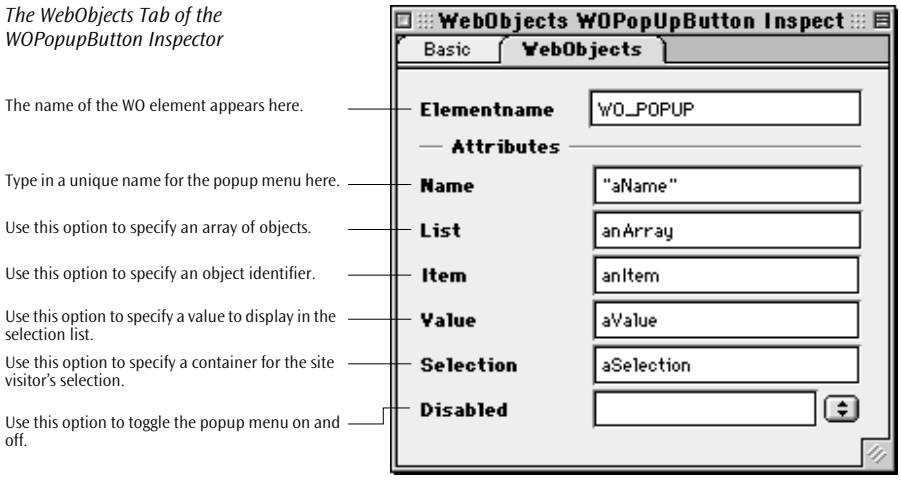
Type in the desired number of rows here.

Check this box to enable multiple selections.

Set the following options in the *Basic* tab of the *WOPopupButton Inspector*:

- In the *Rows* text box, type in the number of rows you want displayed when the site visitor drags the menu.
- Check the *Multiple* checkbox to turn the element into a *WOBrowser* and allow the site visitor to select several options at a time.

The WebObjects Tab of the WOPopupButton Inspector



Set the following options in the *WebObjects* tab of the *WOPopupButton Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *Name* text box to give your WebObjects popup menu a name that uniquely identifies this element within the form. You may specify a name or let WebObjects automatically assign one at runtime.
- Use the *List* text box to specify an array of objects from which the *WOPopupButton* derives its values. For example, *anArray* could name the array containing objects that represent individual items you want the site visitor to choose from.
- Use the *Item* text box to specify an identifier for the elements of the list—for example, *anItem* could represent an object in the *anArray* array.
- Use the *Value* text box to specify a value to display in the selection list—for example, *aValue* for each object in the list.
- *Selection* holds an array of objects that the site visitor can choose from the selection list. For the above example, *Selection* would hold an object from *anArray*. Because a *WOPopupButton* lets the site visitor select only one item at a time, this array holds a single item at a time.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.

WOBrowser

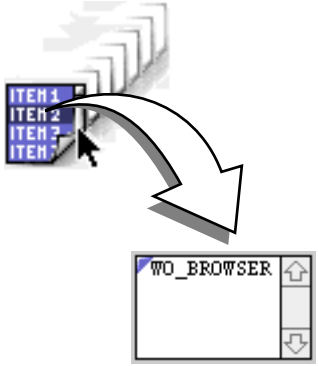
The *WOBrowser* icon inserts a multi-line browser. *WOBrowser* displays itself as a selection list that allows the site visitor to select multiple items at a time.

The related element *WOPopupButton* is similar to *WOBrowser* except that it restricts the site visitor to selecting only one item at a time.

To insert a *WebObjects WOBrowser*, proceed as follows:

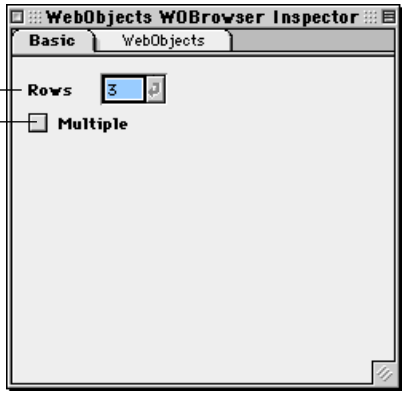
1. Drag the *WOBrowser* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WebObjects Browser



2. Set up the list box in the context-sensitive *Inspector* window, now titled *WOBrowser Inspector*.

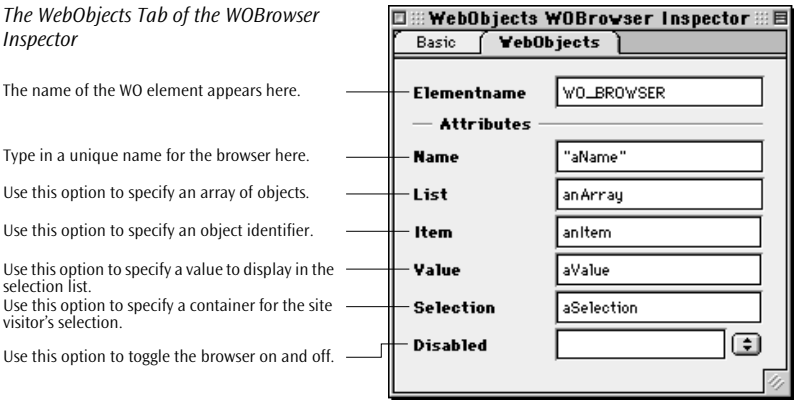
The Basic Tab of the WOBrowser Inspector



Set the following options in the *Basic* tab of the *WOBrowser Inspector*:

- In the *Rows* text box, type in the number of rows you want displayed when the site visitor views the browser.
- Check the *Multiple* checkbox to allow the site visitor to select more than one option at a time.

The WebObjects Tab of the WOBrowser Inspector



Set the following options in the *WebObjects* tab of the *WOBrowser Inspector*:

- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *Name* text box to give your WebObjects browser a name that uniquely identifies this element within the form. You can specify a name or let WebObjects automatically assign one at runtime.
- Use the *List* text box to specify an array of objects from which the *WOBrowser* derives its values. For example, *anArray* could name the array containing objects that represent individual items you want the site visitor to select.
- Use the *Item* text box to specify an identifier for the elements of the list—for example, *anItem* could represent an object in the *anArray* array.
- Use the *Value* text box to specify a value to display in the selection list—for example, *aValue* for each object in the list.
- *Selection* holds an array of objects that the site visitor can choose from the selection list. For the above example, *Selection* would hold one or more objects from *anArray*.
- Use the *Disabled* text box and popup menu to temporarily disable the element. If set to YES, the element appears in the page but is not active. A scripted method can override this property.

WebObjects Header Tags

The *Head Items* view of the *Palette's WebObjects* tab contains a choice of four WebObjects head elements. WebObjects header tags give developers a means to trigger actions. Like their HTML counterparts, WebObjects tags can be inserted in the header section of the Web page to perform specific tasks—for example, load JavaScript code the browser needs to know before loading the body (such as JavaScript dictionaries accessed by scripts throughout the page).

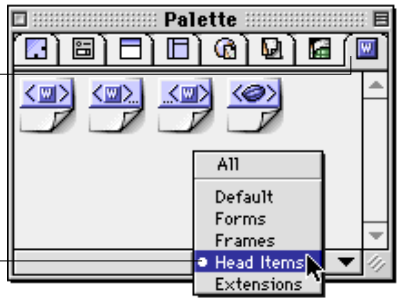
Displaying WebObjects Header Tags

The WebObjects header tag inventory resides in the *Head Items* view of the *WebObjects* tab. To display this view, select *Head Items* from the view control menu at the bottom of the *Palette*.

The WebObjects Tab of the Palette—Head Items View Selected

To view the WebObjects head items inventory, open the WebObjects tab of the Palette, …

… then select Head Items from the Palette's view control menu.



The following pages present the WebObjects head items that the *Palette* offers, sorted in the order of their appearance:

- *WOGenericHeadElement* (see page 100)
- *WOGenericContainer* (see page 101)
- *WOJavaHeadScript* (see page 102)

WOGenericHeadElement

The *WOGenericHeadElement* icon inserts an element that provides a way for WebObjects to accommodate custom empty HTML elements (elements that don't span a range of text) in the header. Because the HTML language is evolving rapidly, it is convenient to have a way to dynamically generate elements that WebObjects does not explicitly support.

To insert a *WOGenericHeadElement* tag, proceed as follows:

1. Drag the *WOGenericHeadElement* tag icon from the *Palette* and drop it in the header section of your page.

Tip: If the header section is not open, drag the item at the triangle control to the left of the small document icon above the main content area, wait for the header section to open, then drag on and drop it in the header window section.

Inserting a WOGenericHeadElement Tag

2 Select the *WOGenericHeadElement* tag to set up the generic element in the context-sensitive *Inspector* window, now titled *WOUnknownType Inspector*. It has four tabs:
   - *Element* lets you set general properties for the current object.
   - *Attributes* lets you edit attributes for the current WebObjects.
   - *Content* displays object-specific text content, if any.
   - *Info* briefly describes the WebObjects element.

3 For instructions on how to set up the new generic element, please refer to the section *WOGenericElement* starting on page 43.

WOGenericContainer

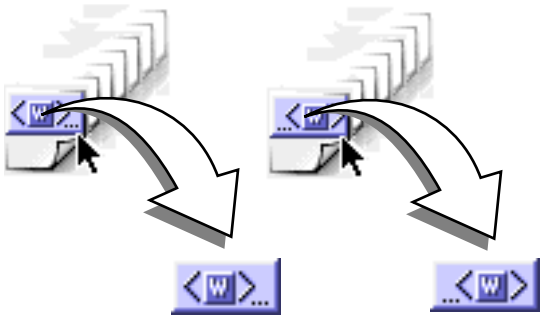
The *WOGenericContainer* icons insert two elements that provide a way for WebObjects to accommodate custom HTML containers (elements that span a range of text) in the header section. Because the HTML language is evolving rapidly, it is convenient to know how to dynamically generate elements that WebObjects does not explicitly support.

To insert a *WOGenericContainer* tag, proceed as follows:

1 Drag the *WOGenericContainer (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window.

2 Drag the *WOGenericContainer (End)* tag icon from the *Palette* and drop it in your layout grid or document window, right behind the *WOGenericContainer (Begin)* tag.

Inserting a WOGenericContainer Tag



3 Select the *WOGenericContainer (Begin)* tag to set up the generic container in the context-sensitive *Inspector* window, now titled *WOGenericContainer Inspector*. It has four tabs:
   - *Element* lets you set general properties for the current WebObjects.
   - *Attributes* lets you edit attributes for the current WebObjects.
   - *Content* displays object-specific text content, if any.
   - *Info* briefly describes the WebObjects element.

4 For instructions on how to set up the new generic container, please refer to the section *WOGenericContainer* starting on page 45.

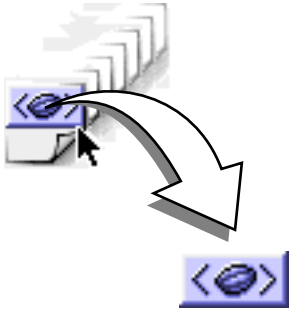
WOJavaHeadScript

The *WOJavaHeadScript* icon inserts an element that lets you embed or write a JavaScript script in the header of a dynamically generated page.

To insert a *WOJavaHeadScript* element, proceed as follows:

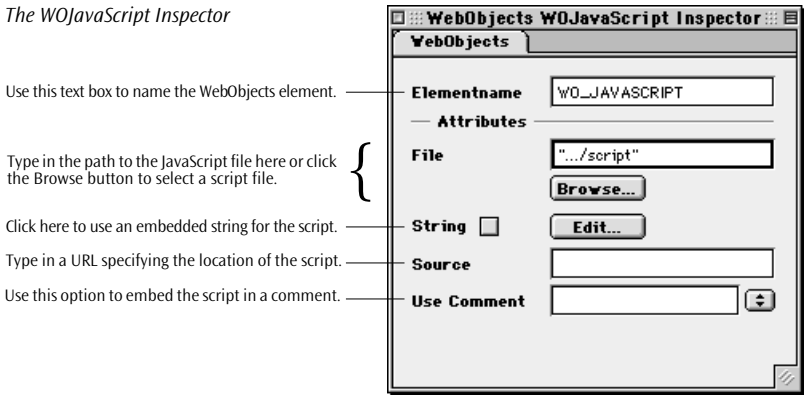
1 Drag the *WOJavaHeadScript* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOJavaHeadScript Placeholder



2 Set up the script in the context-sensitive *Inspector* window, now titled *WOJavaScript Inspector*.

The WOJavaScript Inspector



3 For instructions on how to set up or write a new JavaScript for the header section, please refer to the section *WOJavaScript* starting on page 29.

WebObjects Frames

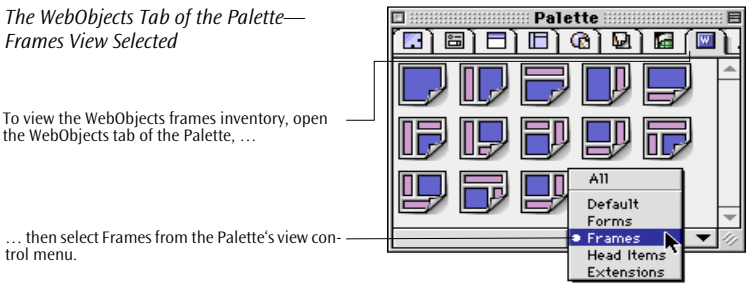
Contained in the *Frames* view of the *WebObjects* tab is a choice of WebObjects frames. Like their HTML counterparts, WebObjects frames let you subdivide the Web page in static sections that can be separately updated, scrolled, or manipulated in any other way. Unlike HTML frames, though, WebObjects frames can be built and filled with content dynamically, depending on input from the visitor.

Frame sets and frames created with any of the *WOFrame*s elements represent themselves as dynamically generated Netscape Frame elements. Either another WebObjects page or a scripted method supports their content.

Displaying WebObjects Frames

The WebObjects frames inventory resides in the *Frames* view of the *WebObjects* tab. To display this view, select the *Frames* item from the view control menu at the bottom of the *Palette*.

The WebObjects Tab of the Palette—Frames View Selected



Using WebObjects Frame Sets and Frames

Inserting Frame Sets

To insert a *WOFrame Set*, proceed as follows:

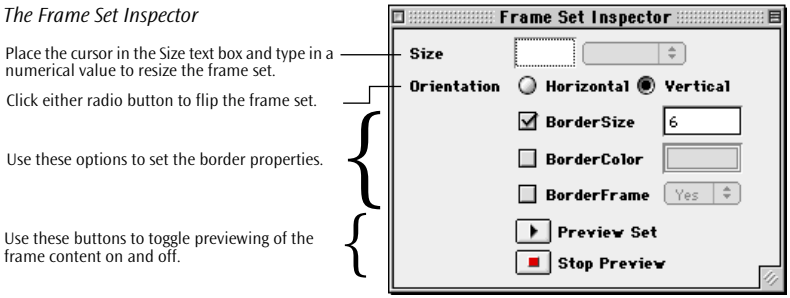
1. Switch to the *Frames* view.
2. Drag any WebObjects frame set from the *Palette* into the document window.

Please note that you can't use HTML frames to build interactive Web pages with WebObjects-based content.

Setting Up a WebObjects Frame Set

If you have some working experience with frames and frame sets, you'll find that the *Frame Set Inspector* offers exactly the same options as its HTML counterpart. The screenshot below introduces the controls of the *Frame Set Inspector*.

The Frame Set Inspector

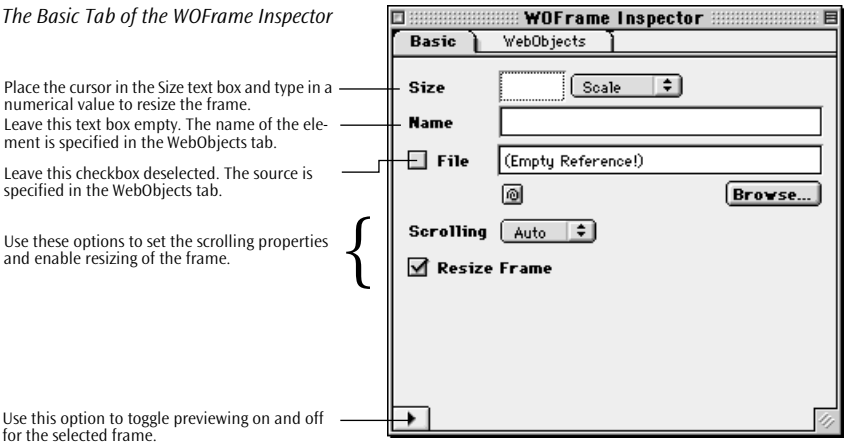


For more detailed instructions, please see Chapter 8, of the *User Guide*.

Setting Up the Basic Properties of WebObjects Frames

The *Basic* tab of the *WOFrame Inspector* lets you adjust basic frame properties, such as size, scrolling behavior, and resize capability.

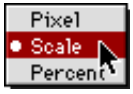
The Basic Tab of the WOFrame Inspector



To set up the basic properties of a *WebObjects* frame in a frame set or document window, proceed as follows:

1. Click the frame to select it and display the *WOFrame Inspector* in the *Inspector* window.
2. Place the cursor in the *Size* text box of the *Frame Inspector* and type in a numerical value to resize the frame.

   The popup menu next to the text box lets you choose the following options:

   - The *Pixel* option lets you enter the width precisely in pixels.
   - The *Scale* option sizes the frame automatically, based on the preferences of the browser. This is the default setting.
   - The *Percent* option lets you enter the size relative to the width of the screen.
3. Place the cursor in the *Name* text box and type in a name for the frame.
4. Leave the *URL* text box empty. The source supplying the frame is selected in the following section.
5. Select an option from the *Scrolling* popup menu:
   - *Auto* enables auto-scrolling, allowing the end user to scroll the page by dragging against the margin of the window.
   - *Yes* enables scrolling via the scrollbar.
   - *No* disables scrolling completely.

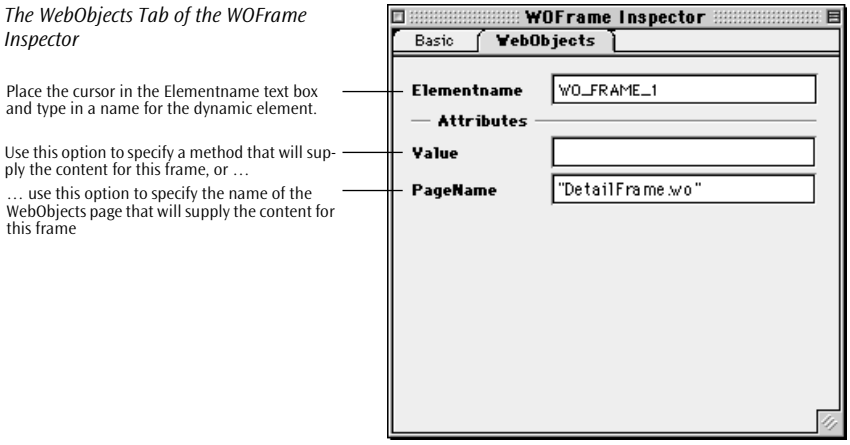
6 Check the *Resize Frame* checkbox to permit resizing of the pane in the Web browser.

7 Click the *Preview Frame* button to toggle previewing on or off for the current frame.

Setting Up the Dynamic Properties of WebObjects Frames

Other than static HTML frames, WebObjects frames need a special setup to account for their dynamic display properties. The major difference is that the content of a frame is provided by the logic acting behind the scenes, rather than by a reference to a static HTML file. You can specify these parameters in the *WebObjects* tab of the *WOFrame Inspector*.

The WebObjects Tab of the WOFrame Inspector



To set up the dynamic properties of a WebObjects frame, proceed as follows:

1 Type in a name for the dynamic element in the *Elementname* text box.

2 Use the *Value* text box to specify a method or the *Pagename* text box to specify the name of the WebObjects component (enclosed in straight quotes) that will supply the content.

WebObjects Preferences

The *WebObjects, Source, Font, Colors,* and *Printing* panes in the *WebObjects* group of the *Preferences* dialog box let you customize various basic settings that influence the behavior and appearance of the WebObjects elements in the source code.

To open the WebObjects group in the *Preferences* dialog box, proceed as follows:

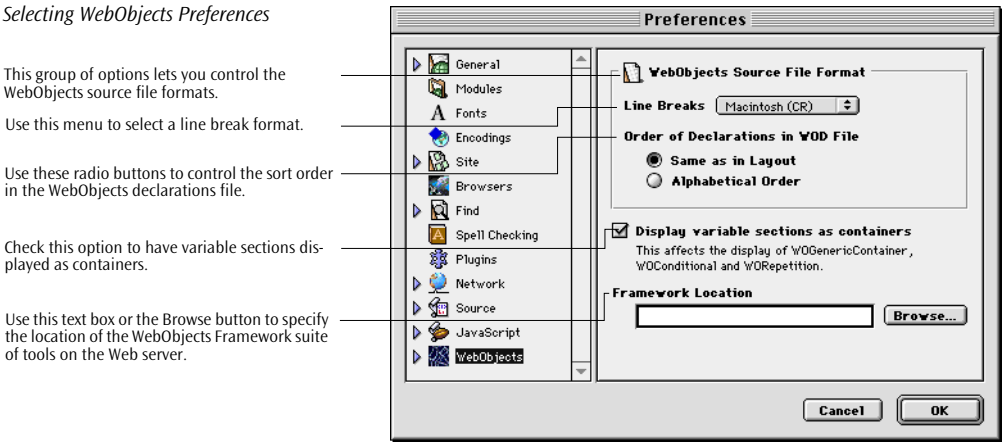
1. Go to the *Edit* menu and choose *Preferences*.
2. Locate and click the *WebObjects* icon in the scrolling sidebar.
3. The WebObjects pane appears (see below).

WebObjects Settings

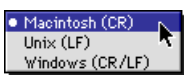
The *WebObjects Preferences* pane lets you control the display of invisible WebObjects elements in the *Layout* view and how to format various items of source code in the source and declaration files.

The WebObjects pane contains the following options:

Selecting WebObjects Preferences



The *WebObjects Source File Format* options let you make two choices:



- Use the *Line Breaks* menu to adjust the line break format that the HTML and declarations files use to the targeted server platform:
  - *Macintosh (CR)* inserts Carriage Return characters only.
  - *Unix (LF)* inserts Line Feed characters only.
  - *Windows (CR/LF)* inserts a Carriage Return/Line Feed character combination.
- The radio buttons in *Order of Declarations in WOD File* (see *The Declaration View* on page 4) control the sort order:

- *Same as in Layout* writes declarations to the wod.file in their order of appearance in the *Layout* view.
- *Alphabetical Order* sorts declarations in the wod.file by alphabet.

- The *Display variable sections as containers* checkbox sets the display mode for WebObjects container elements in *Layout* mode. The example below illustrates how this option influences the display of the *WOConditional* element.

WOConditional Display Modes

or

- Use the *Framework Location* text box or the *Browse* button to specify the location of the WebObjects Framework suite of tools on the Web server.

Source Code Settings

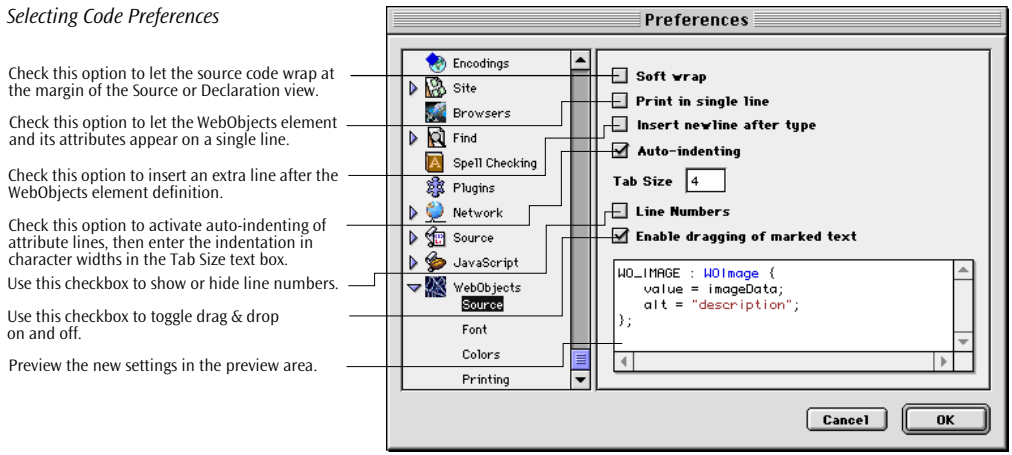
The *Source Preferences* pane lets you control the display of your HTML and WebObjects source code in Adobe GoLive's *Source* and *Declarations* views. A preview area at the bottom of the window lets you see the impact of your changes instantly.

To open the *Source* pane of the *WebObjects Preferences* dialog box, proceed as follows:

1. Go to the *Edit* menu and choose *Preferences*.
2. Locate and click the *WebObjects* icon in the scrolling sidebar.
3. To view more options, click the small triangle control next to the *WebObjects* icon. This expands the *WebObjects* group of preferences and displays four additional options.
4. Click the *Source* item below the *WebObjects* icon.

The *Source* pane contains the following options:

Selecting Code Preferences



- Select the *Soft wrap* option to let the source code wrap at the margin of the *Source* view window.
- Select the *Print in single line* option to have Adobe GoLive write each WebObjects element and its attributes into a single line.

  Example:

```
WO_STRING : WOString { value = anStringObj; };
```

- Select the *Insert newline after type* option to have Adobe GoLive insert an extra line between the WebObjects element and its attributes.

  This option is disabled if the *Print in single line* checkbox is selected.

  Example:

```
WO_STRING : WOString
{
   value = anStringObj;
};
```

- Select the *Auto-indenting* option to have Adobe GoLive indent attributes, then use the *Tab Size* text box to set the width of the indentation (in monospaced character widths).
- Select the *Line Numbers* option to have Adobe GoLive automatically number the lines of WebObjects source code.

Line numbers are a visual feature that the *Source* view supplies. They are not added to the source code.

Example:

```
1 WO_STRING : WOString
2 {
3   value = anStringObj;
4 };
```

- Select the *Enable dragging of marked text* checkbox to activate drag & drop support in the *Source* view.

Font Preferences

The *Font Preferences* pane lets you select a custom font, font size, and font style for your HTML and WebObjects code to override the default *Monaco 9 pt plain* setting.

To open the *Font* pane of the *WebObjects Preferences* dialog box, proceed as follows:

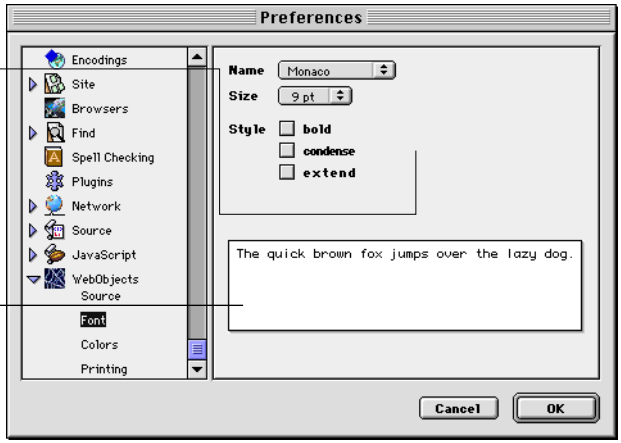
1. Go to the *Edit* menu and choose *Preferences*.
2. Locate and click the *WebObjects* icon in the scrolling sidebar.
3. To view more options, click the small triangle control next to the *WebObjects* icon. This expands the WebObjects group of preferences and displays four additional options.
4. Click the *Font* item below the *WebObjects* icon.

The *Font* pane contains the following options:

Selecting a Custom Font

Use these options to select a custom font and style for plain text displayed in the *Source* and *Declaration* views.

Preview your font preferences here.



Choose an alternative font from the *Name* menu, then set its size and style using the associated font style options.

Color Preferences

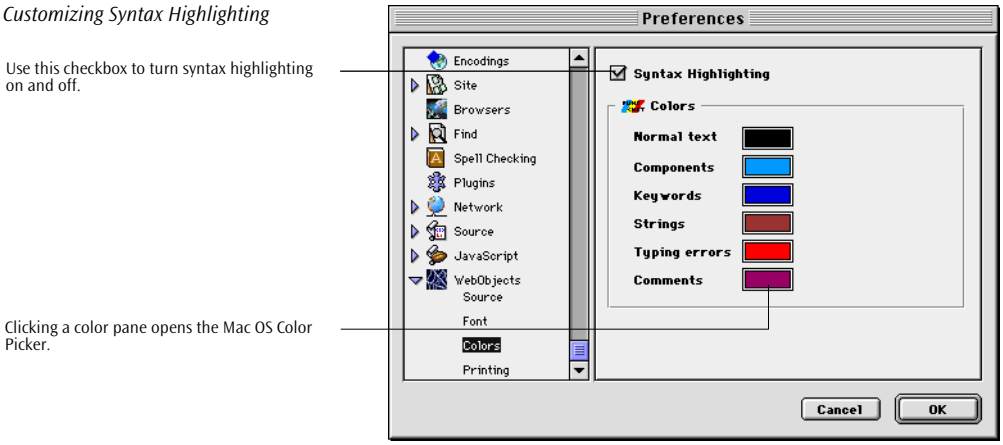
The *Colors* pane lets you enable syntax highlighting for the *Source* and *Declaration* views and select custom colors used for highlighting.

To open the *Colors* pane of the *WebObjects Preferences* dialog box, proceed as follows:

1. Go to the *Edit* menu and choose *Preferences*.
2. Locate and click the *WebObjects* icon in the scrolling sidebar.
3. To view more options, click the small triangle control next to the *WebObjects* icon. This expands the WebObjects group of preferences and displays four additional options.
4. Click the *Colors* item below the *WebObjects* icon.

The *Colors* pane contains the following options:

Customizing Syntax Highlighting



- Select the *Syntax Highlighting* checkbox to activate syntax highlighting.
- Click any of the *Normal text, Components, Keywords, Strings, Typing errors,* and *Comments* color fields to open the Mac OS Color Picker and select display colors.

Printing Preferences

The *Printing* pane lets you control the way your source code is printed in hardcopy and select a custom font, font size, and font style.

To open the *Printing* pane of the *WebObjects Preferences* dialog box, proceed as follows:

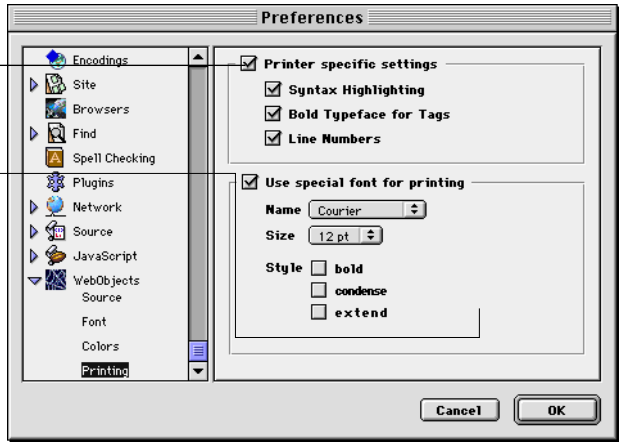
1. Go to the *Edit* menu and choose *Preferences*.
2. Locate and click the *WebObjects* icon in the scrolling sidebar.

3 To view more options, click the small triangle control next to the *WebObjects* icon. This expands the WebObjects group of preferences and displays four additional options.

4 Click the *Printing* item below the *WebObjects* icon.

The *Printing* pane contains the following options:

Setting Printer Preferences



- Select the *Printer specific settings* option to enable the hardcopy printing options.
  - *Syntax highlighting* reproduces the different colors Adobe GoLive uses for highlighting code elements (see *Color Preferences* on page 111) on a color printer.
  - *Bold Typeface for Tags* prints tags in a bold font to make them stand out from the rest of the code.
  - *Line Numbers* adds the optional line numbers (see *Source Code Settings* on page 108) to the printout.
- Select *the Use special font for printing* checkbox and modify the associated style options if you want to use a custom font for printing, instead of the default Geneva 11 pt.

Index