Kapitel 8 WebObjects Support

This chapter describes **GoLive CyberStudio**'s support for WebObjects, a powerful development environment for dynamic Web pages.

General

GoLive CyberStudio comes with an easy-to-use front end for WebObjects, the object-oriented environment from Apple Computer (formerly NeXT Software, Inc.) for developing World Wide Web applications.

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

For example, **GoLive CyberStudio** 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 programmed or scripted by the application developer, this capability gives Web designers a powerful tool for integrating true interactivity.

GoLive CyberStudio's Approach to WebObjects

When writing a WebObjects application, developers create components and connect. A component is a page or portion of a page that has both 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 **GoLive CyberStudio.**
- 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 **GoLive CyberStudio** 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. This script file is usually

written by the application developer using WebScript, a proprietary scripting language for WebObjects.

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

As the preceding list indicates, **GoLive CyberStudio** lets the author generate the "visual" part of a component and declare variables.

Consequently, to use WebObjects you need *WebObjects Enterprise*, the development environment from Apple Computer. For more detailed information on development and deployment licenses, please consult Apple's Web Site at *http://www.enterprise.apple.com*.

Using WebObjects in GoLive CyberStudio

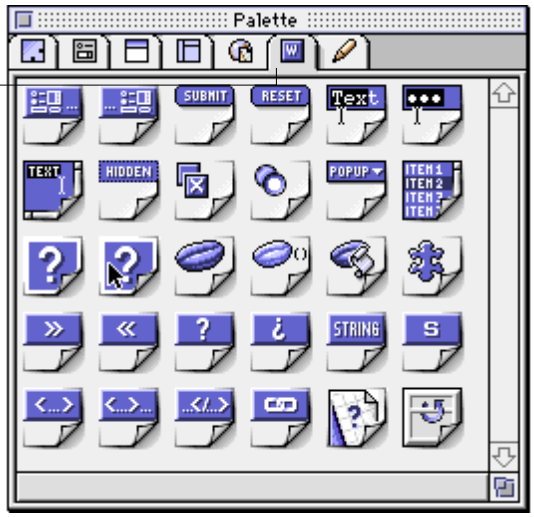
Using WebObjects in **GoLive CyberStudio** lets you build HTML templates with ease—complete with all the features you need to make Web pages interact dynamically with your audience.

WebObjects in the Palette

The *WebObjects* tab of the *Palette* contains a complete selection of building blocks that let you add full interactivity to your pages—with **GoLive CyberStudio**'s known drag & drop ease.

The WebObjects Tab of the Palette

Click at this tab rider to open the WebObjects tab.



See the section *WebObjects Tag Reference* starting on page 227 for a complete reference on the WebObjects elements supported by **GoLive CyberStudio**.

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 required.

The Declaration View

Any entry made in the *Inspector* window is automatically written to the declarations file (*.wod*), which can be viewed by opening the *Declaration* tab of the main document window.

The Declaration View

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

WebObjects and HTML Fragments

WebObjects support requires the use of HTML fragments (see *HTML-Fragmente* on page 63), allowing pages to be dynamically composed of exchangeable portions.

WebObjects and URL Filtering

Using WebObjects elements requires that URL filtering be enabled for all WebObjects script and code files in the *Project Inspector* (see *URL-Filter* on page 419).

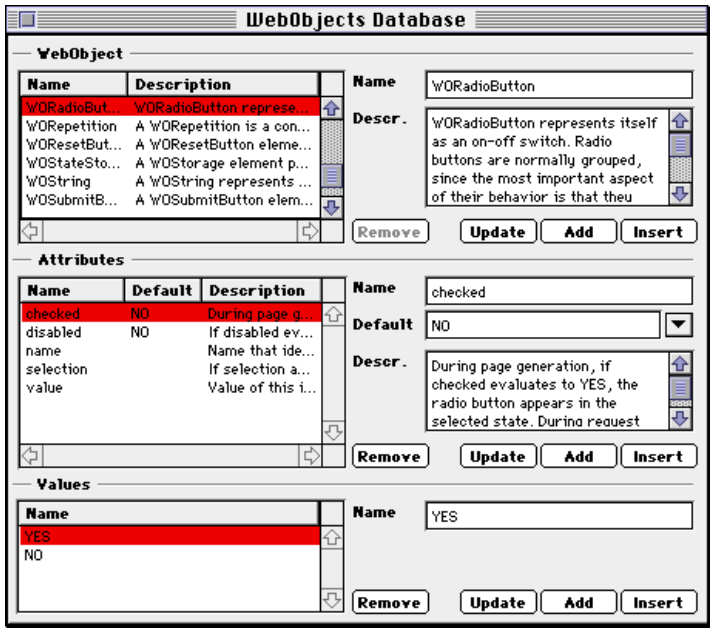
The WebObjects Database

The built-in WebObjects database comes with a complete inventory of WebObjects elements that assists Web authors in choosing the proper building blocks for dynamic pages.

Just as it HTML counterpart, the WebObjects database controls the standard options and default values offered by the element-specific *Inspectors*, thus ensuring consistent use of the WebObjects elements throughout your site.

An example of the WebObjects database editor window appears below.

The WebObjects Database Database Editor Window



Editing the WebObjects Database

Editing the WebObjects database is simple and straightforward. Each window area of the WebObjects database editor contains a list box and task-specific editing tools, allowing the user to add new WebObjects elements, attributes, and values and edit or delete existing entries from the list.

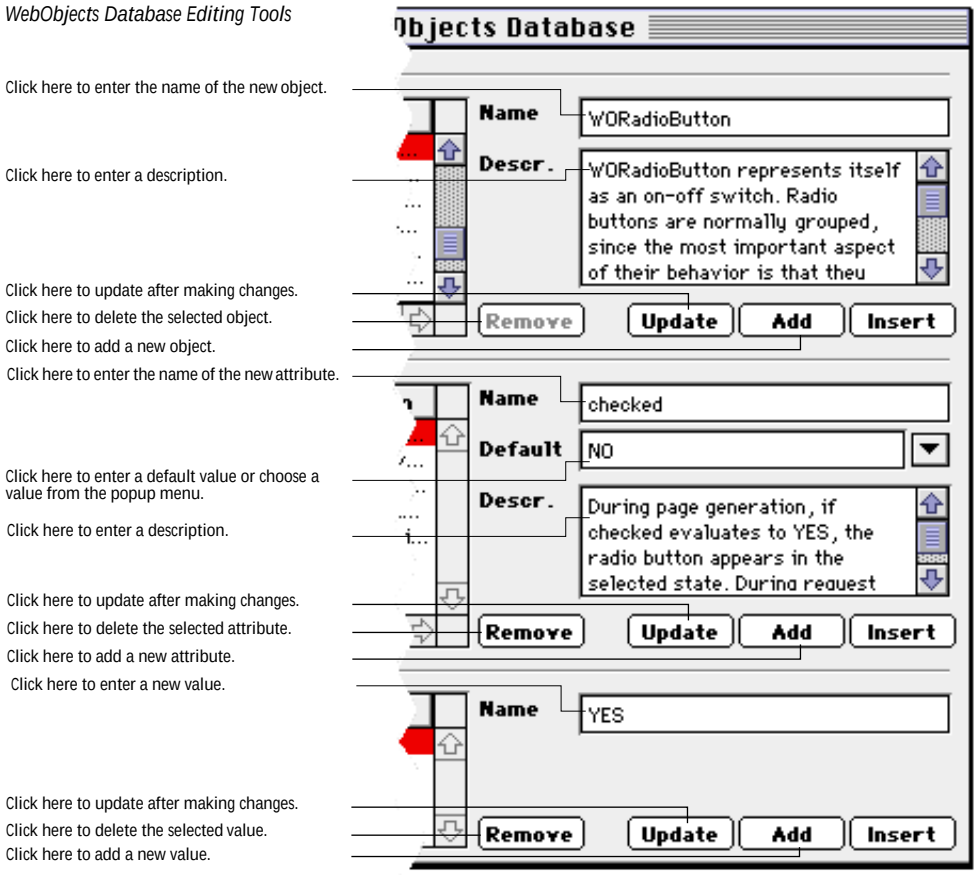
Once stored in the WebObjects database, a user-defined element can be used by inserting a generic WebObject and choosing the proper defintion in the *Inspector* (see *WOGenericElement* on page 283 and *WOGenericContainer* on page 286).

Adding, Updating, or Removing WebObjects Elements

The WebObjects database editor provides a convenient editing environment for adding, updating, or removing WebObjects elements and their attributes and values.

The screenshot below illustrates the editing tools offered by the WebObjects Database and explains their respective functions.

WebObjects Database Editing Tools



Adding a WebObjects Element

To add a WebObjects element, proceed as follows:

1. Place the cursor in the *Name* text box in the *WebObject* section of the database editor window.
2. Enter the name of the new WebObjects element—for example, *WOElement*.
3. Press the Tab key to place the cursor in the *Descr.* text box and enter a descriptive text. This information text will be displayed in the *Info* tab or at another appropriate location in the element-specific *Inspector* window.
4. When you are done entering general properties, click the Add or Insert button to add the new element to the list.
5. Click the Update button to import the new element into the WebObjects database.

Updating

To update a WebObjects element, proceed as follows:

1. Select the WebObjects element to be changed from the *WebObjects* list box.
2. Make the desired changes by editing the name or description.
3. Click the Update button.

Removing

To delete a WebObjects element, proceed as follows:

1. Select the WebObjects element to be deleted from the *WebObjects* list box.
2. Click the Remove button.

Adding an Attribute

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

1. Select the WebObjects element to be changed from the *WebObject* list box.
2. Go to the *Attributes* section of the database editor window and place the cursor in the *Name* text box.
3. Enter the name of the new attribute—for example, *color.*
4. Press the Tab key to place the cursor in the *Default* text box and enter a default value or choose one from the popup menu. For instructions on how to enter new values, see the XXX
5. Press the Tab key to place the cursor in the *Descr.* text box and enter a descriptive text.
6. When you are done adding attribute properties, click the Add or Insert button to add the new attribute to the WebObjects element.

Updating

To update an attribute, proceed as follows:

1. Select the WebObjects element to be changed from the *WebObject* list box.
2. Go to the *Attributes* section of the database editor window and select the attribute to be changed.
3. Make the desired changes by editing the name, value, or description.
4. Click the Update button.

Removing

To delete an attribute, proceed as follows:

1. Select the WebObjects element to be changed from the *WebObject* list box.
2. Go to the *Attributes* section of the database editor window and select the attribute to be deleted.
3. Click the Remove button.

Adding a Value

To add a value to an attribute of a WebObjects element, proceed as follows:

1. Select the element to be changed from the *WebObject* list box.
2. Go to the *Attributes* section of the database editor window and select the attribute you want to add a value to from the *Attributes* list box.
3. Go to the *Values* section of the database editor window and place the cursor in the *Name* text box.
4. Enter the new value—for example, *#F000.*
5. Click the Add or Insert button to add the new value attribute to the current attribute.

   Repeat the steps 4 and 5 above to add more values, if required.

Updating

To update a value, proceed as follows:

1. Select the WebObjects element to be changed from the *WebObject* list box.
2. Go to the *Attributes* section of the database editor window and select the desired attribute from the *Attributes* list box.
3. Go to the *Values* section of the database editor window and select the value to be changed.
4. Edit the value in the *Name* text box.
5. Click the Update button.

Removing

To delete a value, proceed as follows:

1. Select the WebObjects element to be changed from the *WebObject* list box.
2. Go to the *Attributes* section of the database editor window and select the desired attribute from the *Attributes* list box.
3. Go to the *Values* section of the database editor window and select the value to be deleted.
4. Click the Remove button.

WebObjects Tag Reference

The following section lists the WebObjects offered by **GoLive CyberStudio**'s *Palette*.

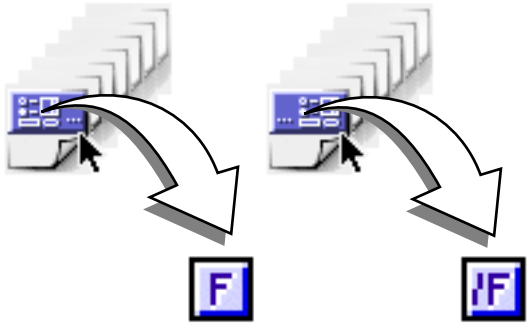
The WOForm Tags

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:

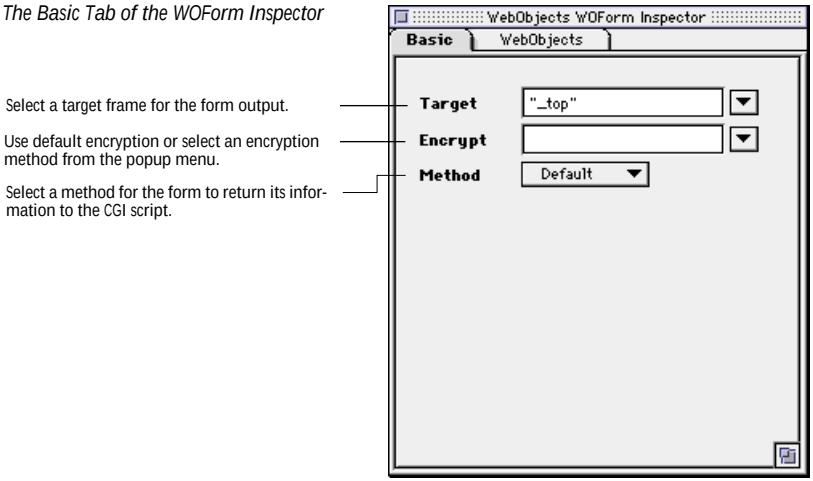
1. Drag the *WOForm (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window to mark the beginning of the form.
2. Drag the *WOForm (End)* tag icon from the *Palette* and drop it in your layout grid or document window to mark the end of the form.

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

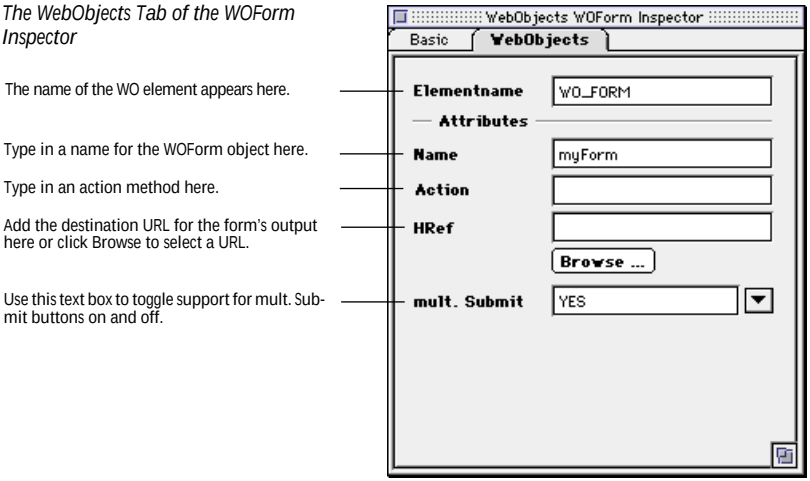


Set the following options in the *Basic* tab of the *WOForm Inspector*:

- 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 the form information shall be sent:
  - *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 a definite length 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 withits own action (such as a WOSubmitButton or a WOActiveImage), that action is invoked instead of the WOForm's.
- 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 *Project* 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.

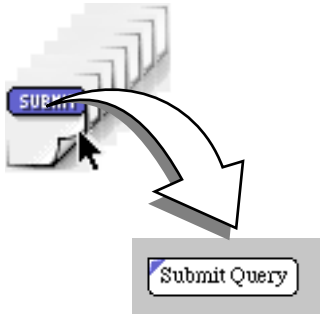
WOSubmitButton

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.

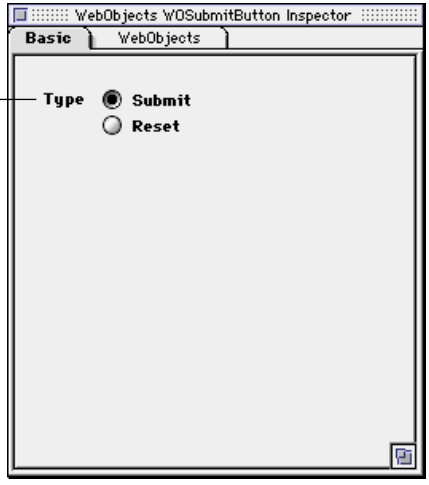
Inserting a WOSubmit Button



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

The Basic Tab of the WOSubmitButton Inspector

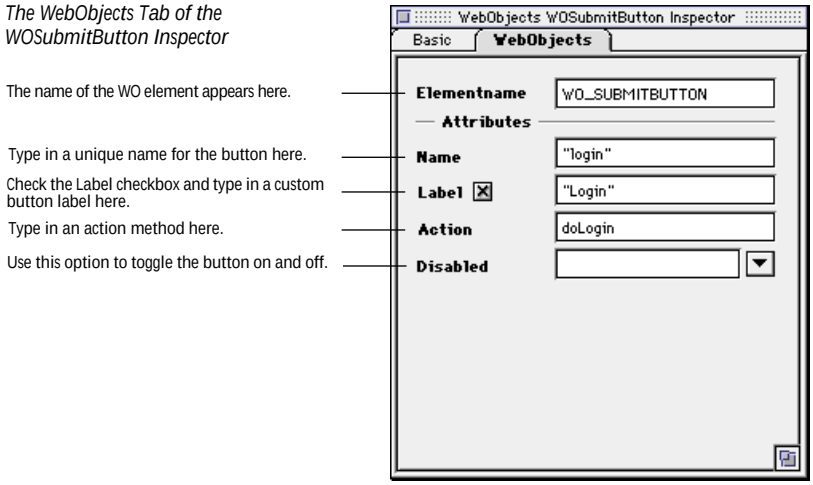
Click the appropriate radio button to toggle the button function.



Set the following options in the *WOSubmitButton Inspector*:

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

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 toggle the element on and off. If set to YES, the element appears in the page but is not active.

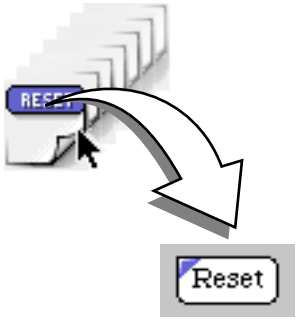
WOResetButton

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.

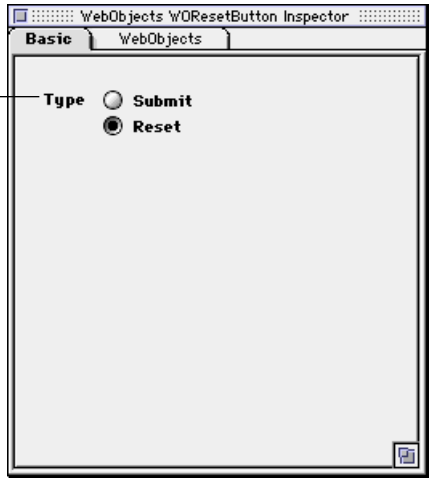
Inserting a WOReset Button



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

The Basic Tab of the WOResetButton Inspector

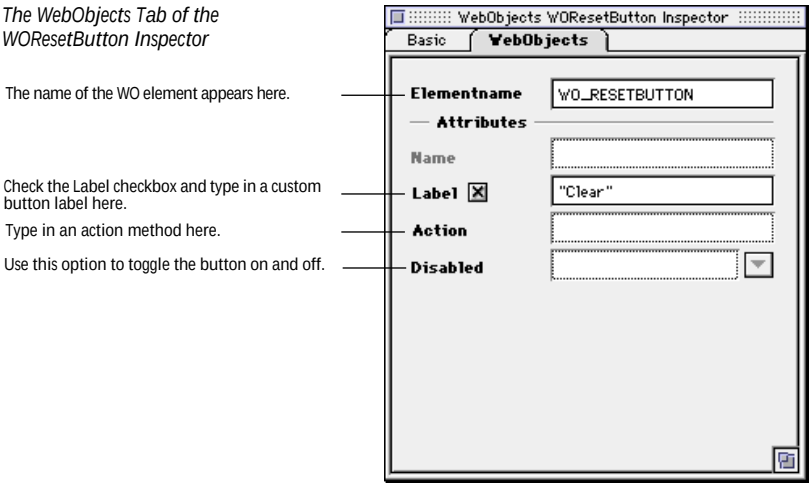
Click the appropriate radio button to toggle the button function.



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



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 toggle the element on and off. If set to YES, the element appears in the page but is not active.

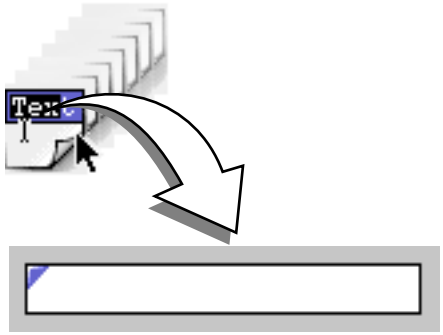
WOTextField

The *WOTextFieId* 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:

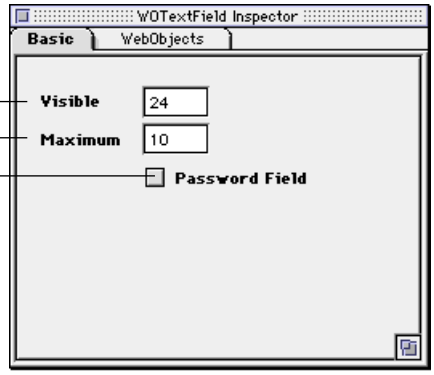
1. Drag the *WOTextFieId* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOTextFieId



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

The Basic Tab of the WOTextFieId Inspector



Type in a numerical value to determine the width of the text box.

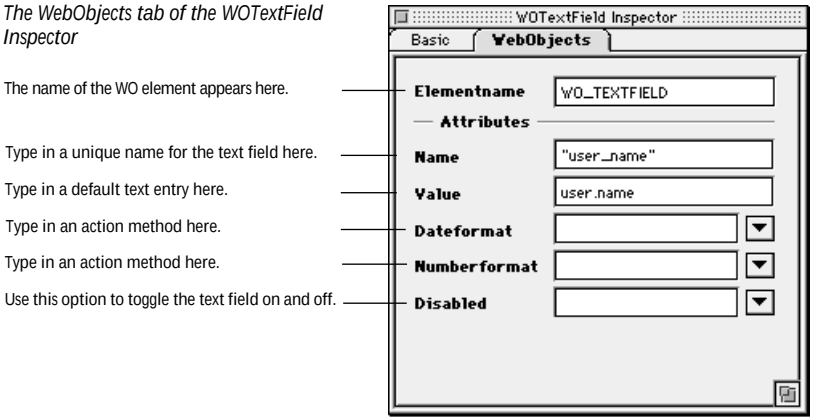
Type in a numerical value to determine the maximum length of the text entry.

Check this checkbox to turn this field into a password field.

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

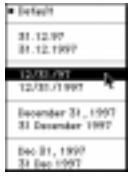
- In the *Visible* text box, type in the number of visible characters to determine the width of the text box.
- In the *Maximum* text box, type in the maximum number of characters accepted by the text box before truncation occurs. If void, this limit is determined by the Web browser used to view the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

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 *Value* should be formatted 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 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 *Value* should be formatted 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 WebObjects documentation for a description of the number format syntax.
- Use the *Disabled* text box and popup menu to toggle the element on and off. If set to YES, the element appears in the page but is not active.

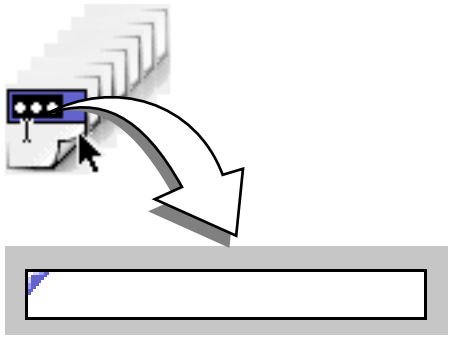
WOPasswordField

The *WOPasswordFieId* 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:

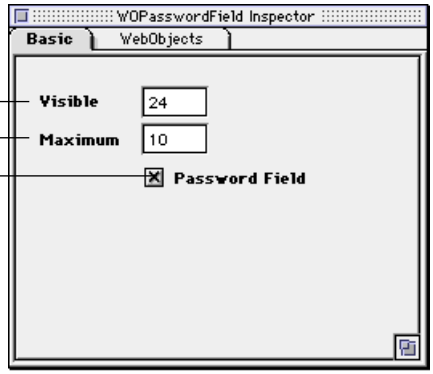
1 Drag the *WOPasswordFieId* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOPasswordFieId



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

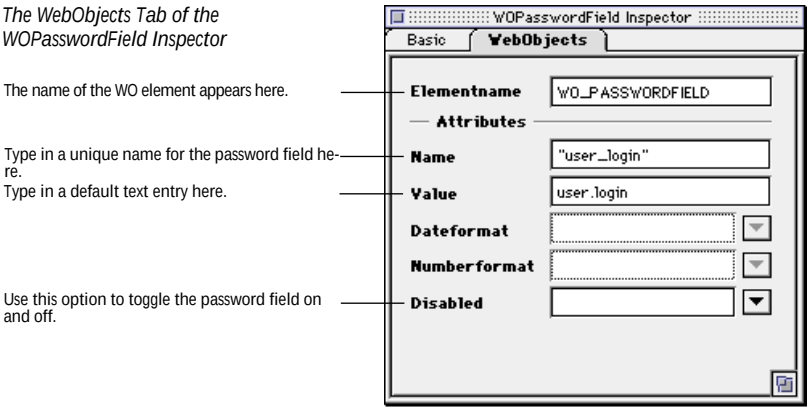
The Basic Tab of the WOPasswordFieId Inspector



Set the following options in the *Basic* tab of the *WOPasswordFieId 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 accepted by the text box before truncation occurs. If void, 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 *WOPasswordField 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 may 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 toggle the element on and off. If set to YES, the element appears in the page but is not active.

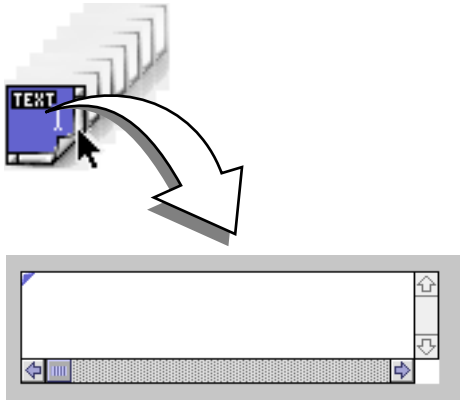
WOText

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.

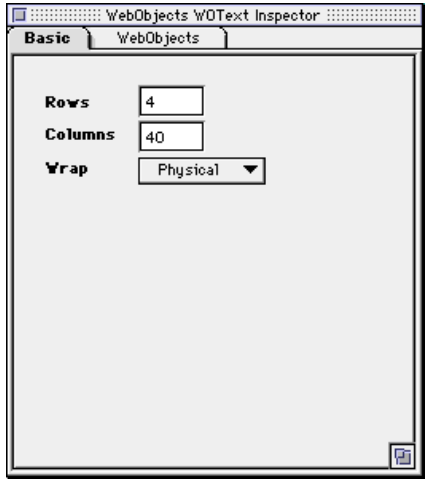
Inserting a WOText



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

The WOText Inspector

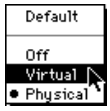
Use these options to resize the WOText element and control the behavior of the text.



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 the default settings for *WOText* objects.
  - *Wrap* 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

The name of the WO element appears here.

Type in a unique name for the WOText field here.

Type in a default text entry here.

Use this option to toggle the WOText field on and off.

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 may 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 toggle the element on and off. If set to YES, the element appears in the page but is not active.

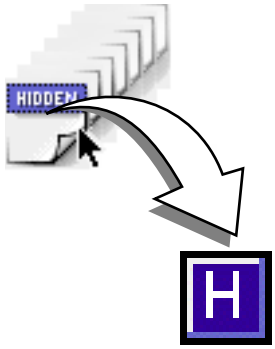
WOHiddenField

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 280) element is designed expressly for this purpose.

To insert a hidden tag, proceed as follows:

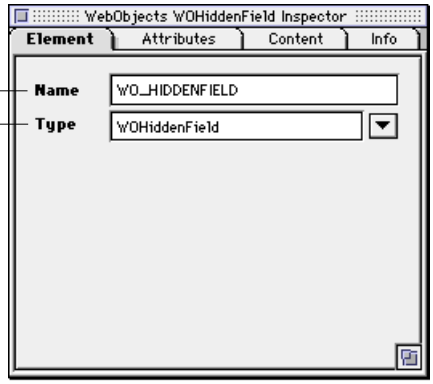
1. Drag the *WOHiddenField* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOHiddenField Tag



2. Set up the *WOHiddenField* tag in the context-sensitive *Inspector* window, now titled *WOHiddenField Inspector*. The *WOHiddenField Inspector* has four tabs:
   - *Element* lets you set general properties for the current WebObject.
   - *Attributes* lets you edit attributes for the current WebObject.
   - *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.

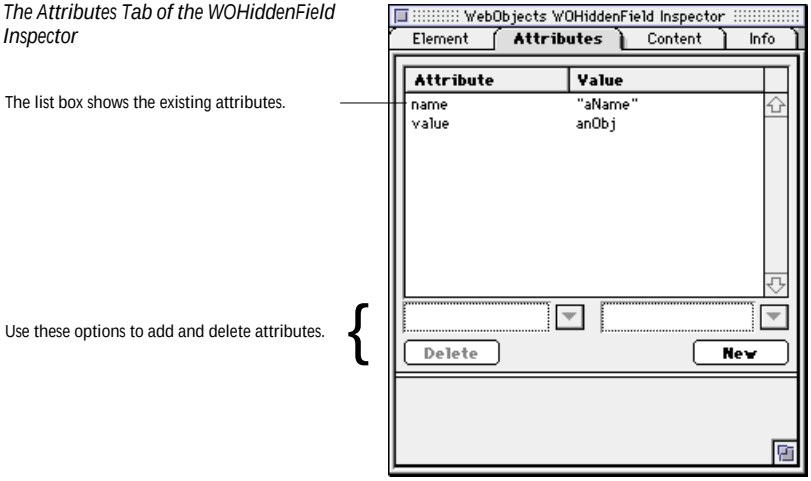
”

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 may 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 WebObject, if required.

Caution: Use the *Type* option with care. Your settings may be lost after changing the type of an object!

The Attributes Tab of the WOHiddenField Inspector



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 have **GoLive CyberStudio** add the new attribute to the list.
- 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.

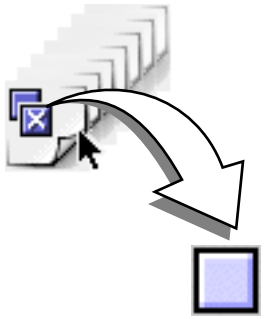
WOCheckbox

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:

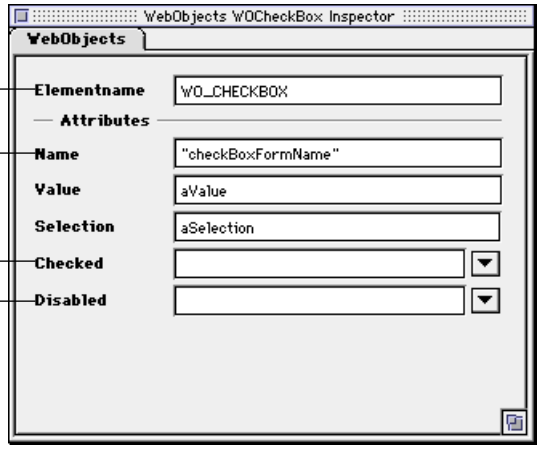
1 Drag the *WOCheckbox* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOCheckbox



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

The WOCheckbox Inspector



The name of the WO element appears here.

Type in a unique name for the checkbox here.

Use these options to specifiy a default value and control the behavior of the checkbox at runtime.

Use this option to set a default state for the checkbox.

Use this option to toggle the checkbox on and off.

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 checkbox 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 check box in: YES if checked; NO if not.
- Use the *Disabled* text box and popup menu to toggle the element on and off. If set to YES, the element appears in the page but is not active.

WORadioButton

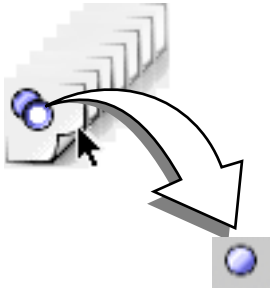
The *WORadioButton* icon inserts a radio button that acts as an on-off switch. Radio buttons are normally grouped because their main purpose is to allow the site visitor to select exactly one of several choices. If the site visitor selects one button, the previously selected button is deselected.

Because radio buttons usually appear as a group, *WORadioButton* is commonly placed within a *WORepetition* (see page 271).

To insert a radio button, proceed as follows:

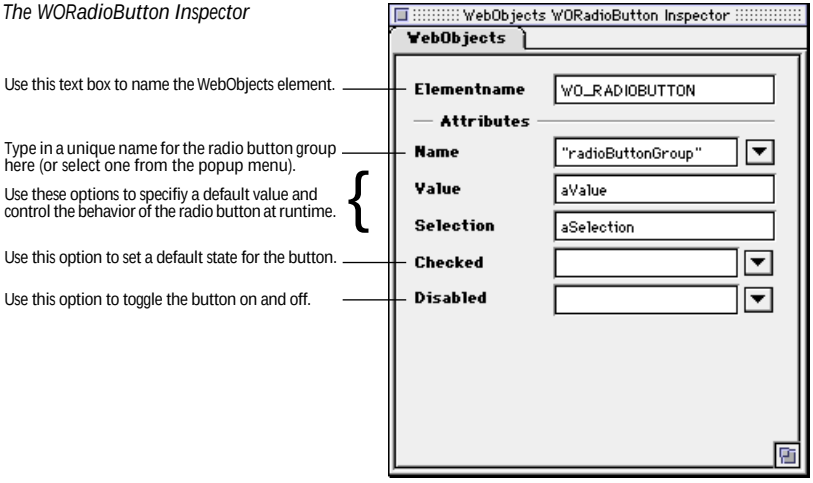
1 Drag the *WORadioButton* icon from the *Palette* and drop it in your layout grid or document window.

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. Only one radio button at a time can be selected within a group.
- 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 select 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 toggle the element on and off. If set to YES, the element appears in the page but is not active.

Note that either *Checked* or *Value* is required in a WORadioButton declaration, but that they are mutuallyexclusive.

WOPopupButton

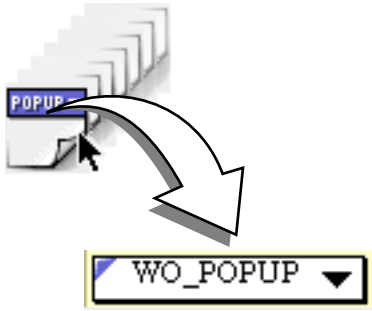
The *WOPopupButton* icon inserts a popup menu with multiple options to choose from. *WOPopupButton* displays itself as a selection list that allows the site visitor to select only one item at a time.

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:

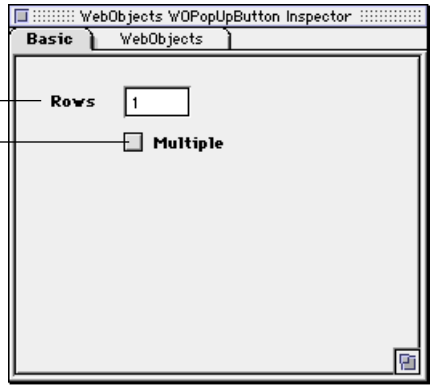
1. Drag the *WOPopupButton* icon from the *Palette* and drop it in your layout grid or document window.

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



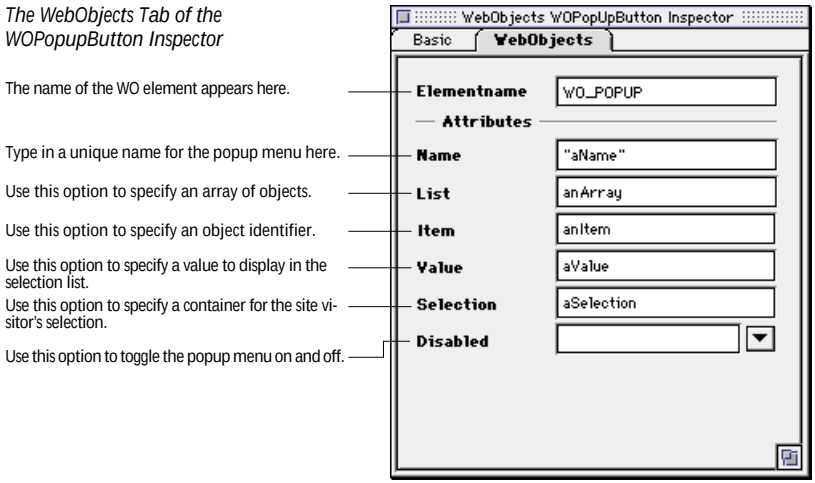
Type in the desired number of rows here.

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 allow the site visitor to select more than one option 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 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 chose from the selection list. For the above example, selection would hold an object from *anArray*. Since a WOPopupButton lets the site visitor select only one item at a time, this array holds justone item at a time.
- Use the *Disabled* text box and popup menu to toggle the element on and off. If set to YES, the element appears in the page but is not active.

WOBrowser

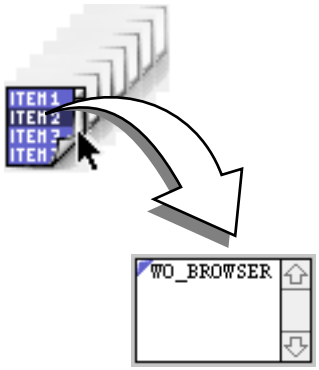
The *WOBrowser* icon inserts a multi-line browser. *WOBrowser* displays itself as a selection list that allows the site visitor to select multiple items at a time.

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 Browser*, proceed as follows:

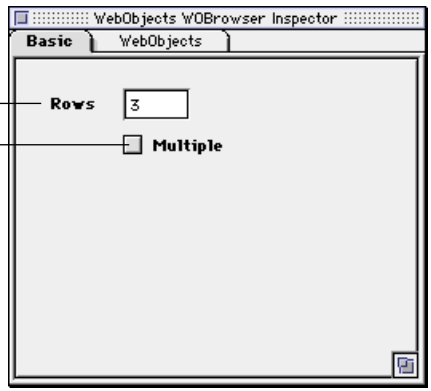
1 Drag the *WOBrowser* icon from the *Palette* and drop it in your layout grid or document window.

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



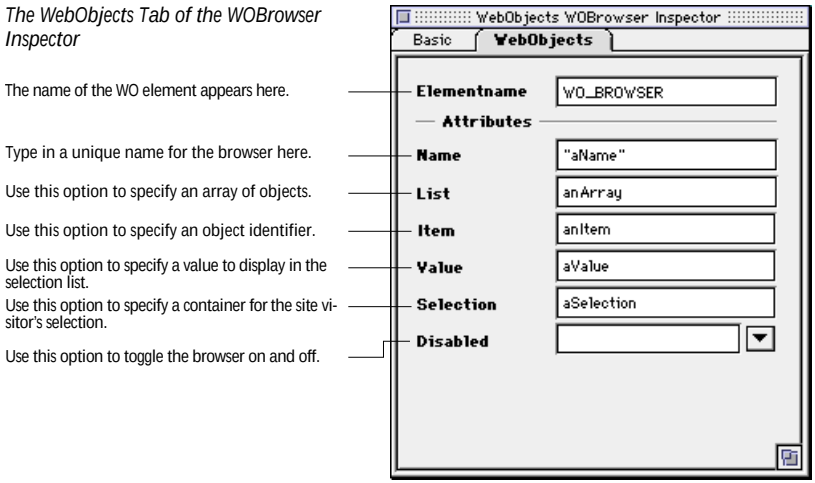
Type in the desired number of rows here.

Check this box to enable multiple selections.

Set the following options in the *Basic* tab of the *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 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 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 chose 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 toggle the element on and off. If set to YES, the element appears in the page but is not active.

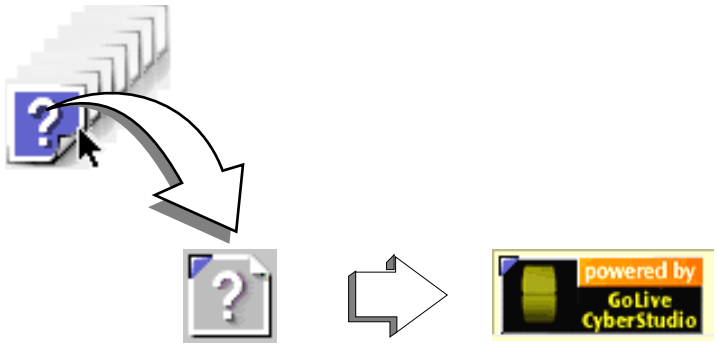
WOImage

The *WOImage* icon inserts an image placeholder that can be dynamically or statically linked with an image. Images help the audience to navigate your page—for example, by clicking buttons.

To insert a WOImage placeholder, proceed as follows:

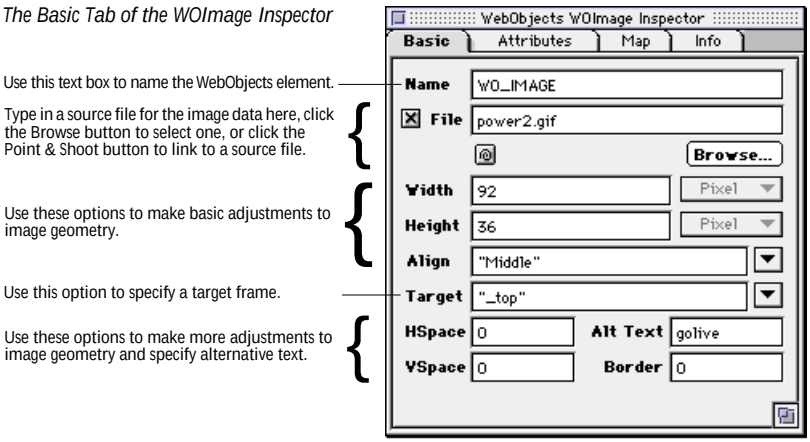
1. Drag the *WOImage* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOImage Placeholder



2. Set up the image in the context-sensitive *Inspector* window, now titled *WOImage Inspector*. The *WOImage Inspector* 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* 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*:

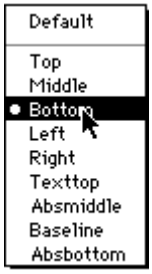
- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *File* text box and check box combination to type in a path to the file containing the image data. The source file can be statically specified in the declarations file or it can be dynamically specified using 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 (formerly 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 *Project* 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 in the horizontal spacing in pixels and press the Return key to confirm your entry.
2. In the *VSpace* text box, type in 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:

- 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 have **GoLive CyberStudio** 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.

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.

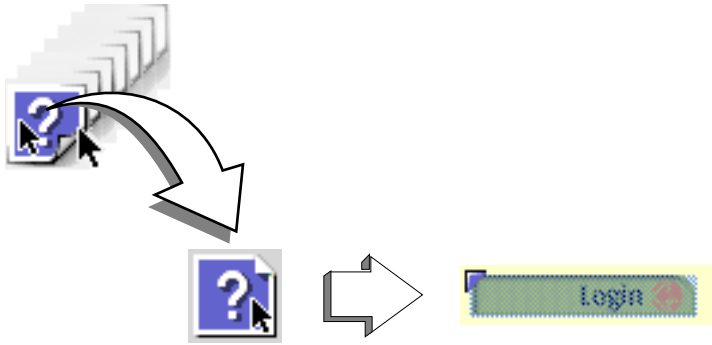
If located outside an HTML form, a WOActiveImage functions as a mapped, active image. When the site visitor clicks such a WOActiveImage, the coordinates of the click are sent back to the server. Depending on where the site visitor clicks, different actions can be invoked. An image map file associates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

Within an HTML form, a WOActiveImage functions as a graphical submit button. You typically use WOActiveImages 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.

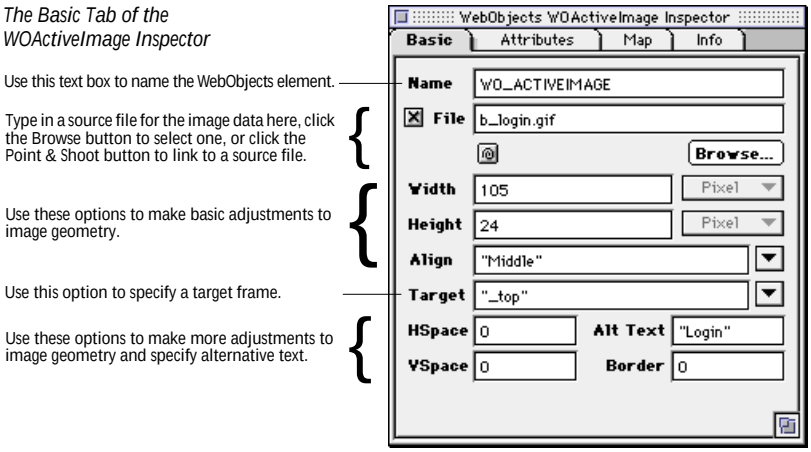
Inserting a WOActiveImage Placeholder



2 Set up the image in the context-sensitive *Inspector* window, now titled *WOActiveImage Inspector.* The *WOActiveImage Inspector* 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 for map creating server-side clickable image maps.
- *Info* briefly describes the WebObjects element.

The Basic Tab of the WOActiveImage Inspector



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

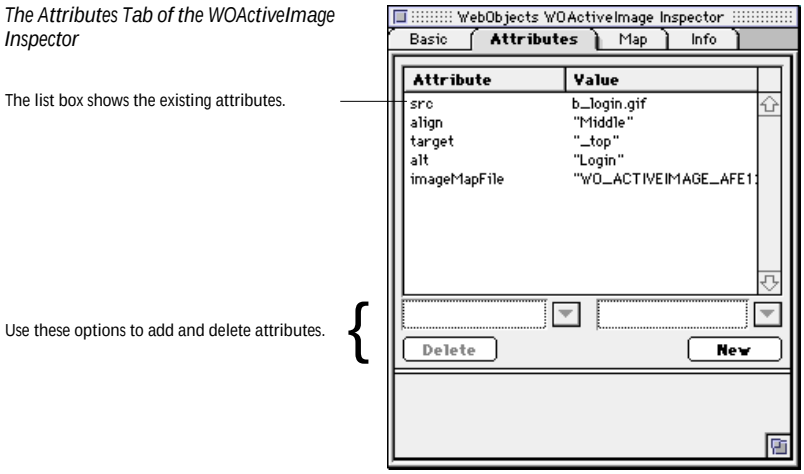
- The *Elementname* text box displays the name of the current WebObjects element.
- Use the *File* text box and check box combination to type in a path to the file containing the image data. The source file can be statically specified in the declarations file or it can be dynamically specified using 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 Enterprise Software (formerly 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 *Project* window.
- Use the *Target* popup menu to choose a frame in a frameset 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 250
- *Adjusting Vertical and Horizontal Spacing* on page 252
- *Entering Alternative Text* on page 253
- *Adjusting Border Width* on page 253

The Attributes Tab of the WOActiveImage Inspector



The *Attributes* tab of the *WOActiveImage 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:

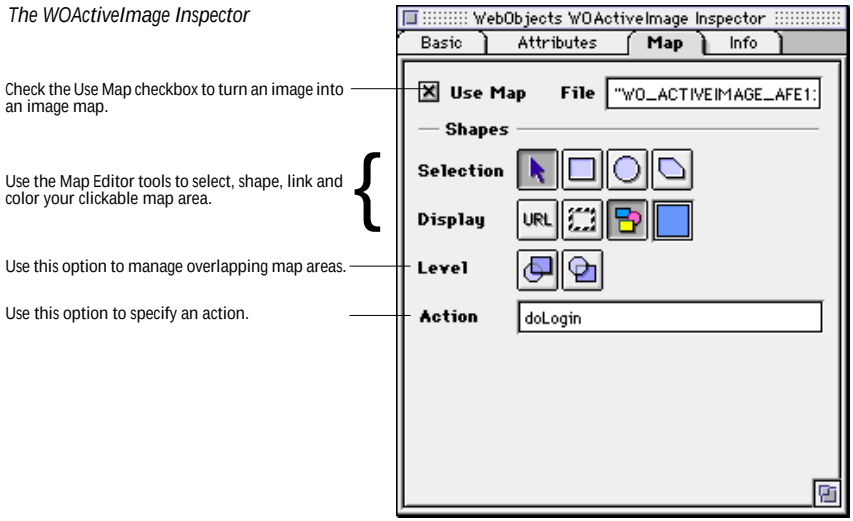
- Click the New button to add new attributes for the *WOActive-Image*. Clicking the New button enables the two text boxes below the list box, allowing you to type in the attribute name (left) and value (right). Press Return to confirm your entry and have **GoLive CyberStudio** add the new attribute to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

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

GoLive CyberStudio 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 in no time at all. 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 is passed to the WebObjects development environment for further processing.

The WOActiveImage Inspector



WOActiveImage objects must be located outside an HTML form— that is, they may not be enclosed in WOForm tags (see page 227)—to function as active images.

Also, please note that the *Use Map* checkbox will be 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 on to your document window.
2. With the *WOActiveImage* selected, click the *Map* tab rider in the *WOActiveImage Inspector* window.
3. In the *Map* tab, check the *Use Map* checkbox.
4. **GoLive CyberStudio** will create an image map file now. The filename, consisting of the WebObjects element name, a hexadecimal identifier, and the suffix *.map*, will appear in the *File* text box of the *Map* tab rider.
5. Place the cursor in the *Action* text box and type in the name of a method or script specifying the action to be taken 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 rider. 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:

7 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.

WebObjects does not support polygons yet. In future releases, this button will allow you to draw polygonal hot spots.

8 Draw the map at the desired location.

9 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 Picker*, allowing you to select a different fill color. (Red is default.)

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

10 If you are working with multiple overlapping hot spots, click the following buttons to manage:

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.

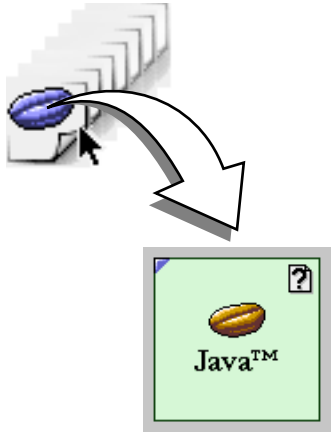
WOApplet

The *WOApplet* icon inserts a dynamic element that generates HTML to specify a Java applet. The applet's parameters are passed by one or more *WOParam* elements.

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.

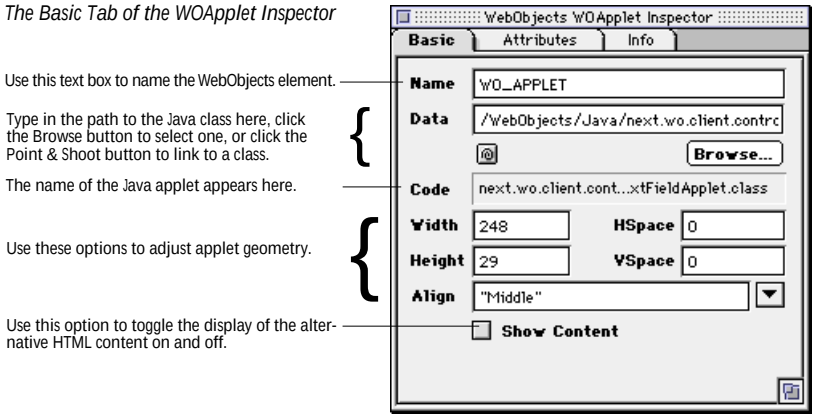
Inserting a WOApplet Placeholder



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

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

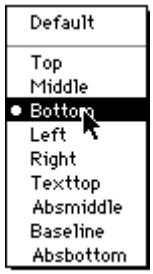
The Basic Tab of the WOApplet Inspector



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

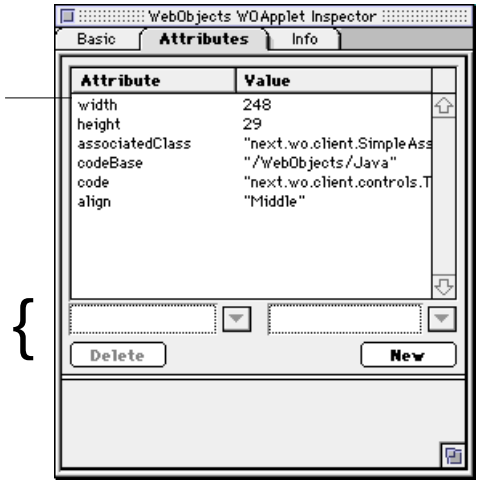
- Use the *Name* text box to give the WebObjects element a unique name. This identifies the element as a unique entity in case there are corresponding applets on the same page.
- Use the *Data* text box to type in the path to the Java applet. If the directory containing the applet code is omitted, the 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 *Project* 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 in pixels of the area to allocate 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 in pixels of the area to allocate 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 folllowing options:
  - The *DefaulI* 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.

The Attributes Tab of the WOApplet Inspector



The list box shows the existing attributes.

Use these options to add and delete attributes.

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:

- 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 have **GoLive CyberStudio** add the new attribute to the list.

- 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.

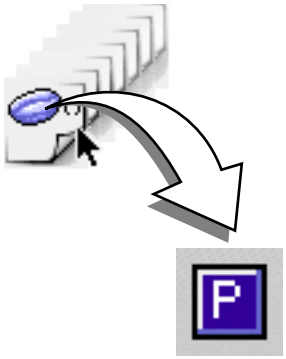
WOParam

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:

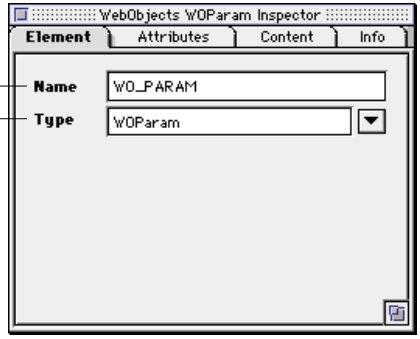
1. Drag the *WOParam* icon from the *Palette* and drop it on the *WOApplet* placeholder on your layout grid or in the document window.

Inserting a WOParam Placeholder



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

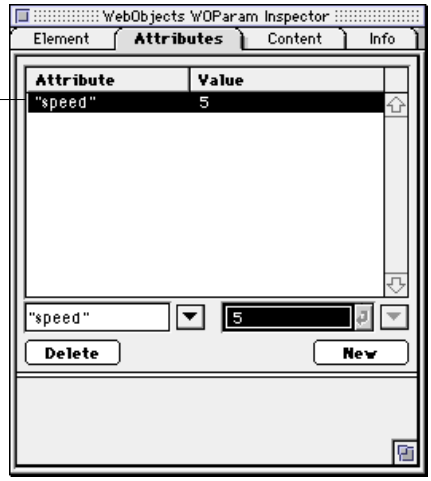
The Element Tab of the WOParam Inspector



Set the following options in the *Basic* 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 WebObject, if required.

The Attributes Tab of the WOApplet 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:

- 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 have **GoLive CyberStudio** 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.

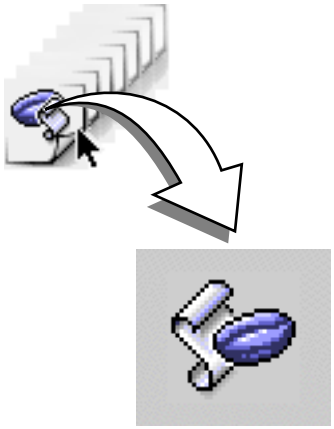
WOJavaScript

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:

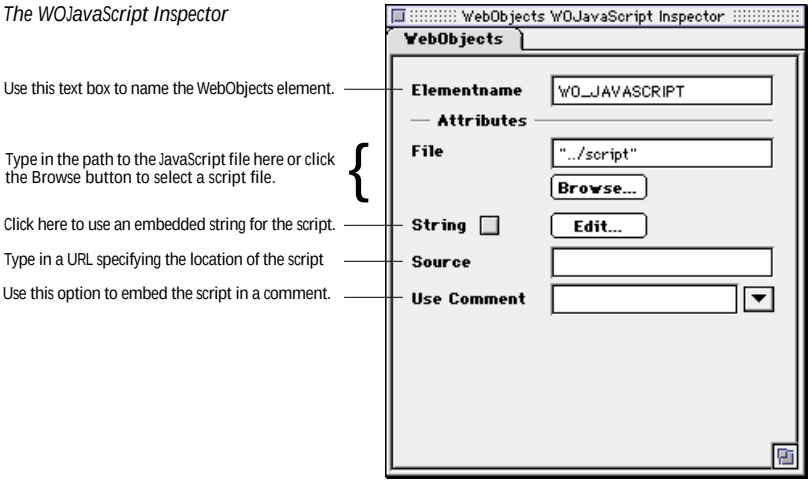
1 Drag the *WOJavaScript* icon from the *Palette* and drop it in your layout grid or document window.

Inserting a WOJavaScript Placeholder



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

The WOJavaScript Inspector



Set the following options in the *WOJavaScript Inspector*:

- Use the *Elementame* 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 Enterprise Software (formerly NeXT Software Inc.).
- Use the *String* checkbox and Edit button to embed a string containing the script in the HTML page. 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.

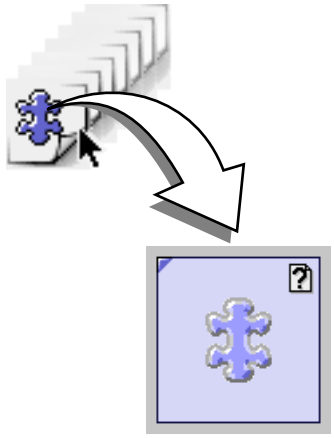
WOEmbeddedObject

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.

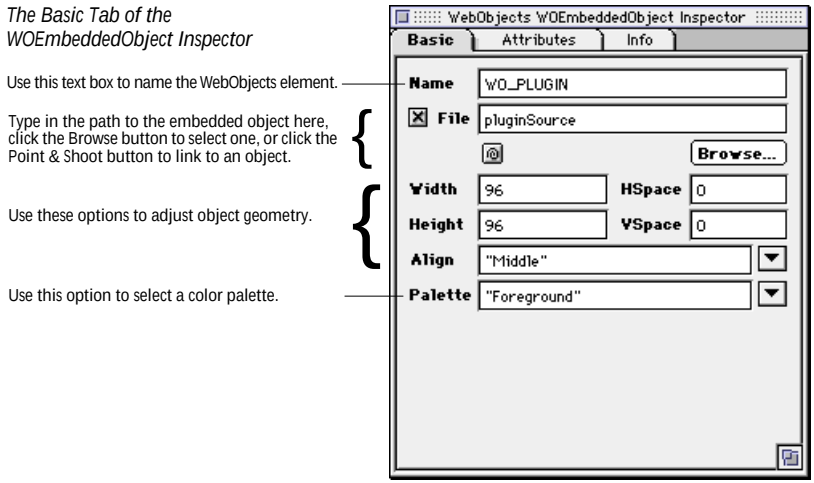
Inserting a WOEmbeddedObject Placeholder



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

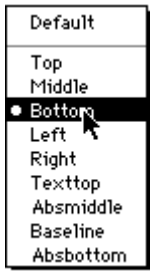
The Basic Tab of the WOEmbeddedObject Inspector



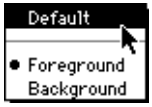
Set the following options in the *Basic* tab of the *WOEmbeddedObject Inspector*:

- Use the *Name* text box to give the WebObjects element a unique name.
- Use the *File* text box and check box 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 *Project* window.
- Click to place the cursor in the *Width* text box, andenter the desired overall width in pixels of the area to allocate 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 in pixels of the area to allocate 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.

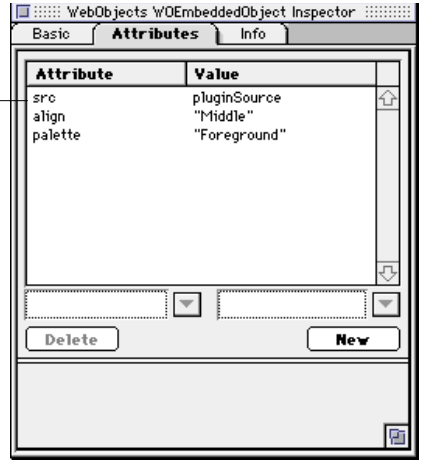


- Use the *Align* text box and popup menu to align the plugin relative to text on the same line. The *Align* popup menu lets you select the folllowing options:
  - The *Default* option uses the alignment settings of the surrounding text.
  - The *Top* option aligns surrounding text with the top of the 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 list box shows the existing attributes.

Use these options to add and delete attributes.

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:

- 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 have **GoLive CyberStudio** add the new attribute to the list.
- 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.

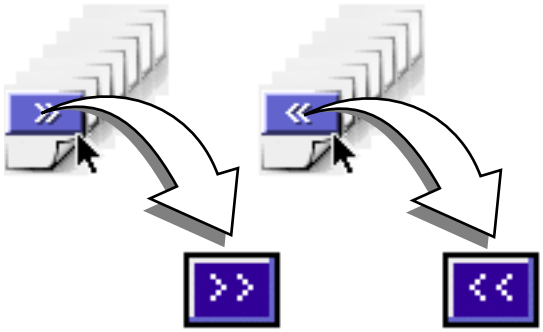
WORepetition

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:

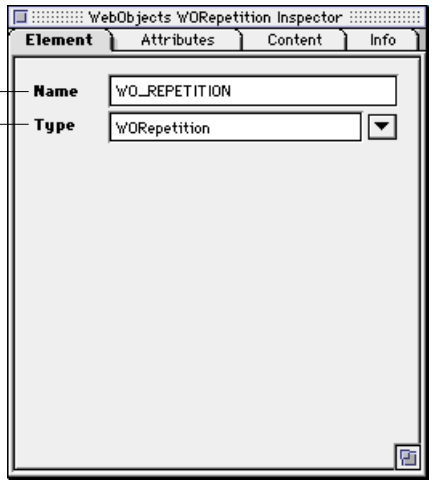
1. Drag the *WORepetition (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window, right in front of the item(s) to be repeated.
2. Drag the *WORepetition (End)* tag icon from the *Palette* and drop it in your layout grid or document window, right behind the item(s) to be repeated.

Inserting a WORepetition Tag



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

The Element Tab of the WORepetition Inspector



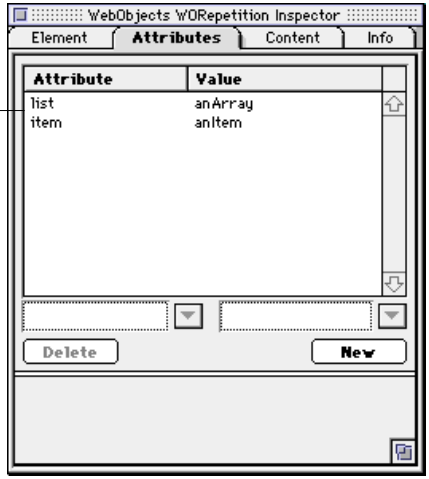
Use this text box to name the WebObjects element.

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 WebObject, if required.

The Attributes Tab of the WORepetition Inspector



The list box shows the existing attributes.

Use these options to add and delete attributes.

The *Attributes* tab of the *WORepetition 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:

- 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 have **GoLive CyberStudio** add the new attribute to the list.
- Currently supported attributes include:
  - *list*: Array of objects that the WORepetition will iterate through
  - *item:* Current item in the list array
  - *index:* Index of the current iteration of the WORepetition
  - *identifier:* Value used to uniquely identify this item in the list array. Typically it is the primary key of an enterprise object
  - *count:* Number of times this element will repeat its contents
- 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.

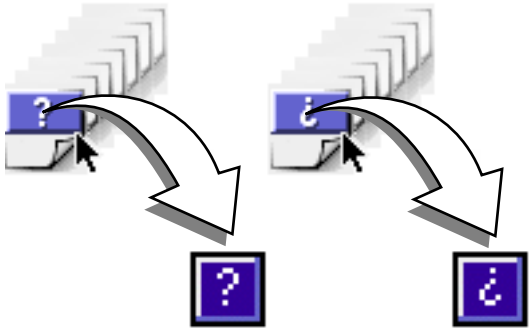
WOConditional

The *WOConditional* icons insert a conditional 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:

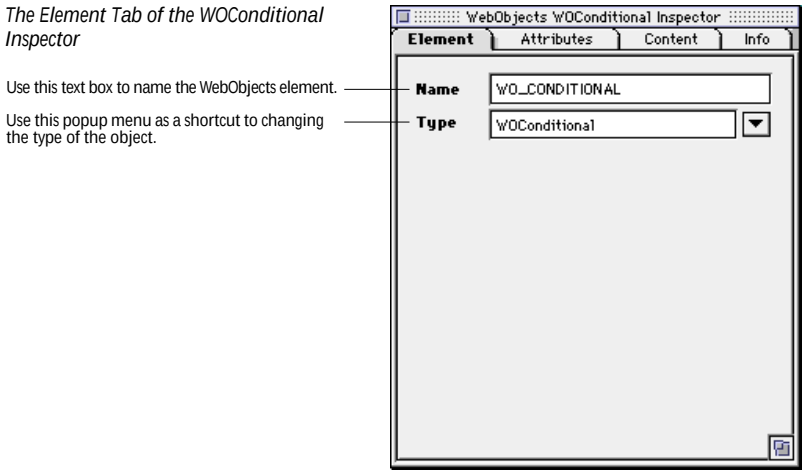
1. Drag the *WOConditional (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window, right in front of the item(s) you want to be conditional.
2. Drag the *WOConditional (End)* tag icon from the *Palette* and drop it in your layout grid or document window, right behind the item(s) you want to be conditional.

Inserting a WOConditional Tag



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

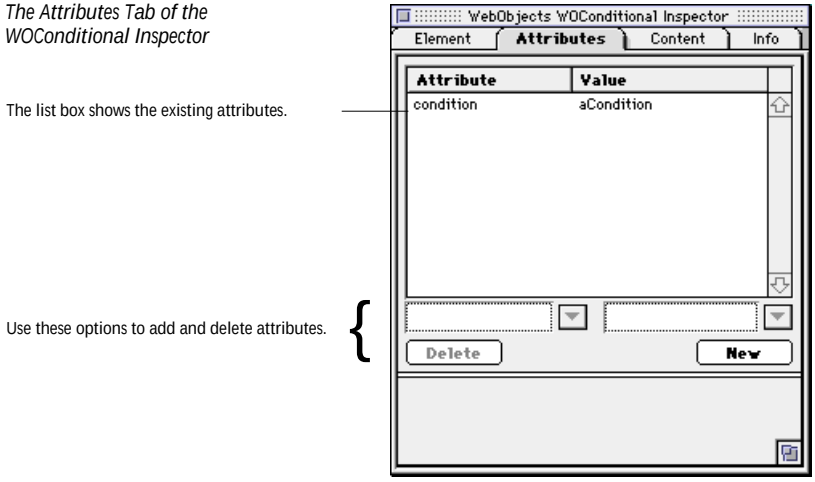
The Element Tab of the WOConditional Inspector



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 WebObject, if required.

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:

- Click the New button to add new attributes for the *WOConditional* 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 have **GoLive CyberStudio** add the new attribute to the list.
- Currently supported attributes include:
  - *condition*: Expression to evaluate. If the expression evaluates to YES (assuming negate is NO), the HTML code controlled by the WOConditional object is emitted; otherwise it is not.
  - *negate*: Inverts the sense of the condition. By default, negate is assumed to be NO.
- 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.

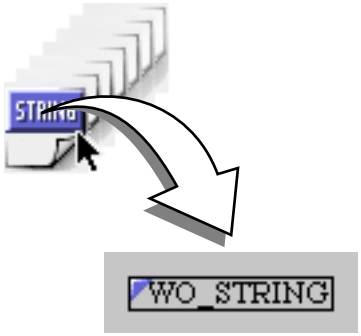
WOString

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:

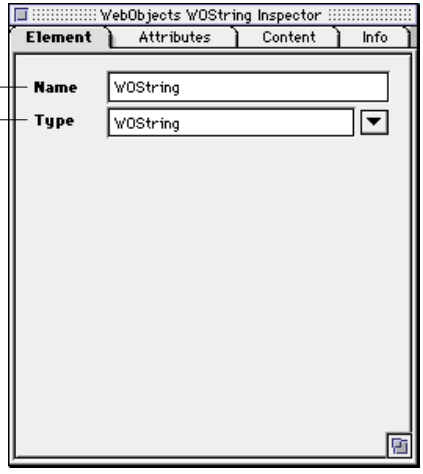
1. Drag the *WOString* tag icon from the *Palette* and drop it in your layout grid or document window.

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.* The *WOString Inspector* has four tabs:
   - *Element* lets you set general properties for the current WebObject.
   - *Attributes* lets you edit attributes for the current WebObject.
   - *Content* displays object-specific text content, if any.
   - *Info* briefly describes the WebObjects element.

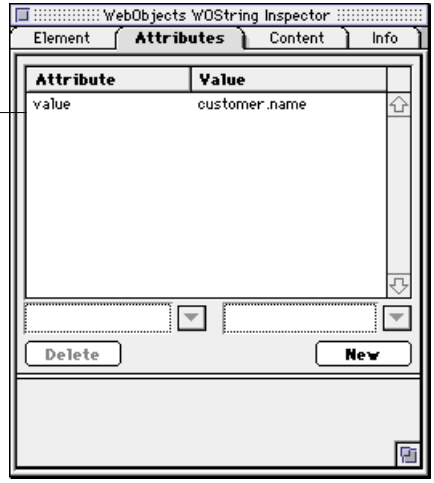
The Element Tab of the WOString Inspector



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 WebObject, if required.

The Attributes Tab of the WOString Inspector



The list box shows the existing attributes.

Use these options to add and delete attributes.

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

To add and delete attributes, proceed as follows:

- 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 have **GoLive CyberStudio** 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 occupied by this dynamic element. For a description of the NSString class, please refer to the documentation on foundation classes, available from Apple Computer (formerly NeXT Software, Inc.)

- *dateformat*: This format string specifies how *value* should be formatted 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 *value* should be formatted 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 escapeHTML is set to NO, WebObjects simply passes the string to the browser without protecting HTML tags from being interpreted as commands.

• 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.

WOStateStorage

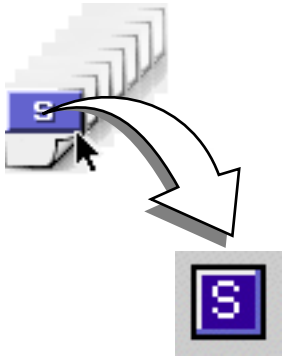
The *WOStateStorage* icon inserts an element that provides a simple mechanism for storing application state in an HTML page. If you include a *WOStateStorage* element in a form, any session and persistent data will be stored in the page rather than on the server.

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.

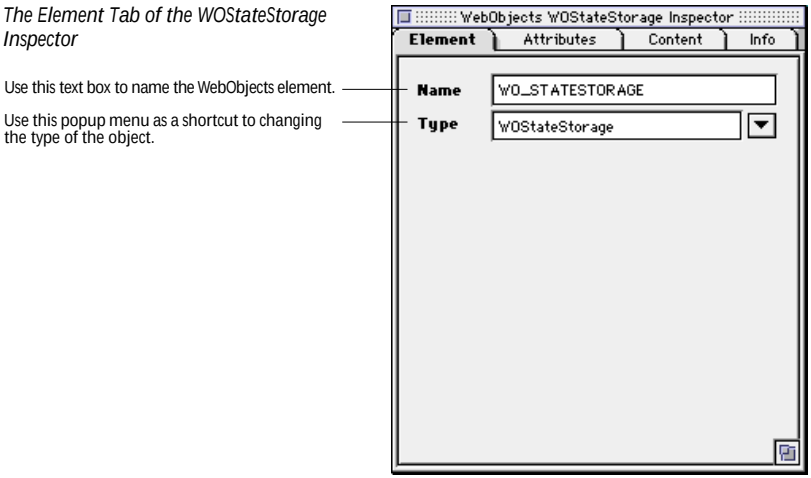
Inserting a WOStateStorage Tag



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

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

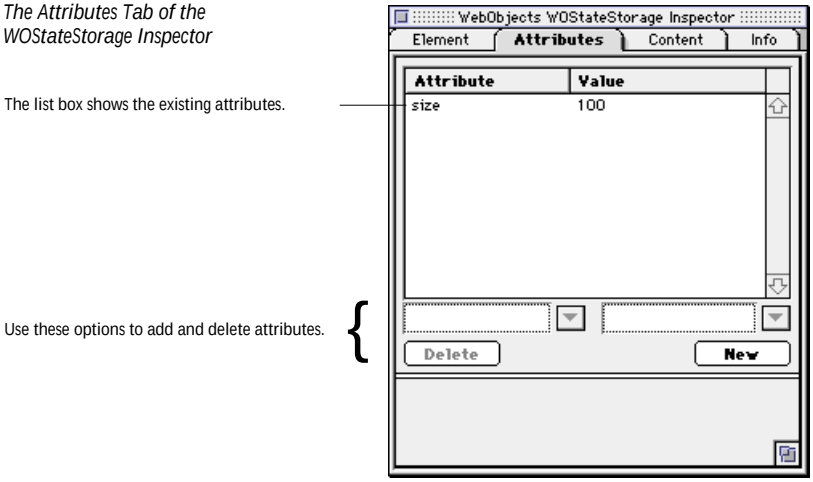
The Element Tab of the WOStateStorage Inspector



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 WebObject, 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:

- 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 have **GoLive CyberStudio** 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.
- 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.

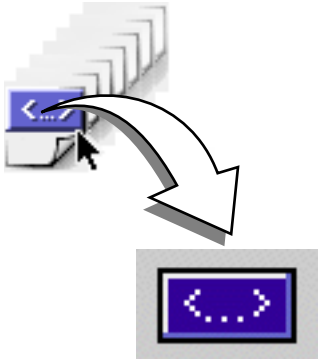
WOGenericElement

The *WOGenericElement* icon inserts an element that provides a way for WebObjects to accommodate custom HTML tags that are empty— that is, that 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:

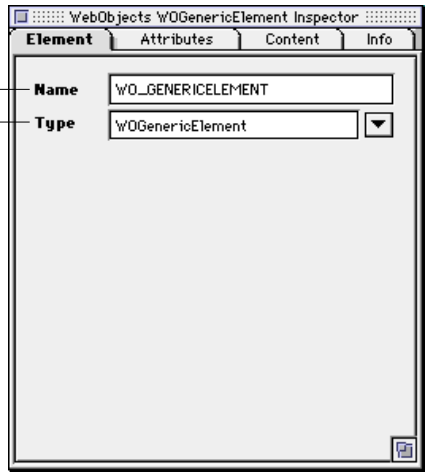
1. Drag the *WOGenericElement* tag icon from the *Palette* and drop it in your layout grid or document window.

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.* The *WOGenericElement Inspector* has four tabs:
   - *Element* lets you set general properties for the current WebObject.
   - *Attributes* lets you edit attributes for the current WebObject.
   - *Content* displays object-specific text content, if any.
   - *Info* briefly describes the WebObjects element.

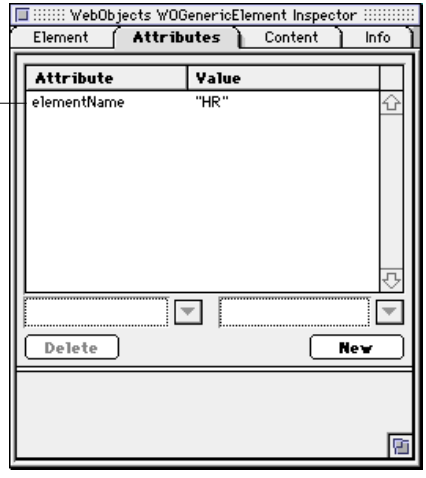
The Element Tab of the WOGenericElement Inspector



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 Database* on page 222).

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:

- Click the New button to add new attributes for the *WOGenericElement*. 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 have **GoLive CyberStudio** add the new attribute to the list.
- Currently supported attributes include:
  - *elementName*: This is the name of the HTML element to generate. This name (for example, “HR”) will be 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 is in the HTML document.
- 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.

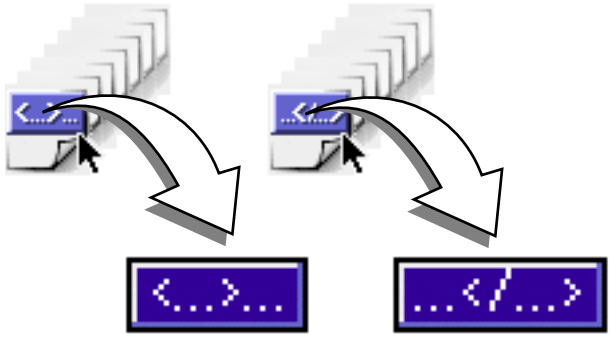
WOGenericContainer

The *WOGenericContainer* icons insert two elements that provide 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 are not explicitly supported by WebObjects.

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

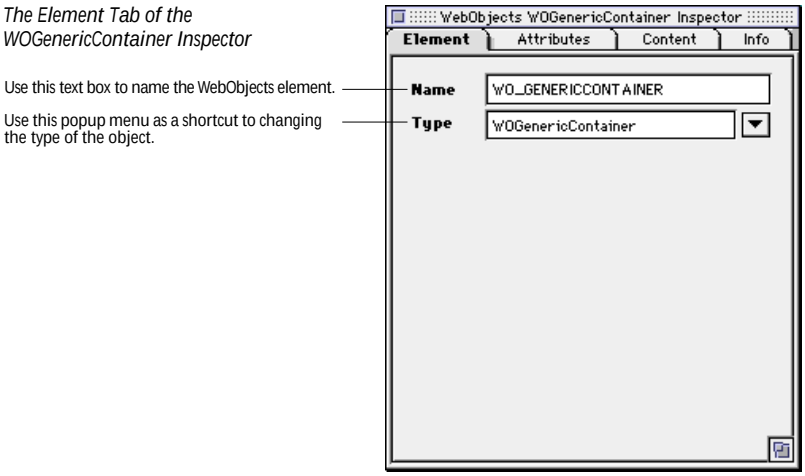
1. Drag the *WOGenericContainer (Begin)* tag icon from the *Palette* and drop it in your layout grid or document window.
2. Drag the *WOGenericContainer (End)* tag icon from the *Palette* and drop it in your layout grid or document window, right behind the *WOGenericContainer (Begin)* tag.

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.* The *WOGenericContainer Inspector* has four tabs:
   - *Element* lets you set general properties for the current WebObject.
   - *Attributes* lets you edit attributes for the current WebObject.
   - *Content* displays object-specific text content, if any.
   - *Info* briefly describes the WebObjects element.

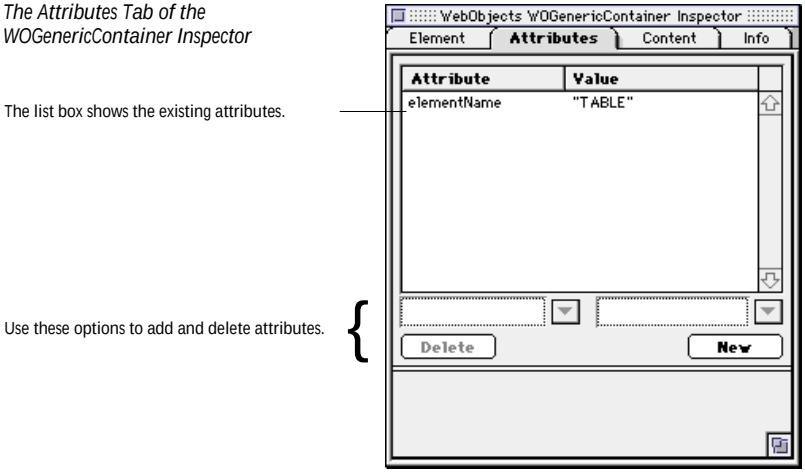
The Element Tab of the WOGenericContainer Inspector



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 required.

To add and delete attributes, proceed as follows:

- 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 have **GoLive CyberStudio** 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”) will be used to generate the container’s opening and closing tags. *elementName* must be statically defined as a constant. It can not 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.
- 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.

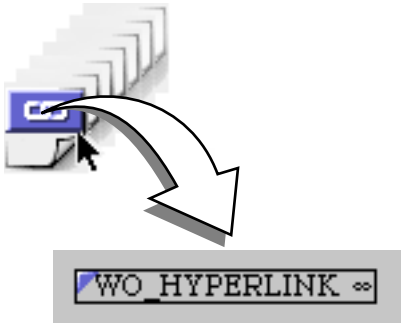
WOHyperlink

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.

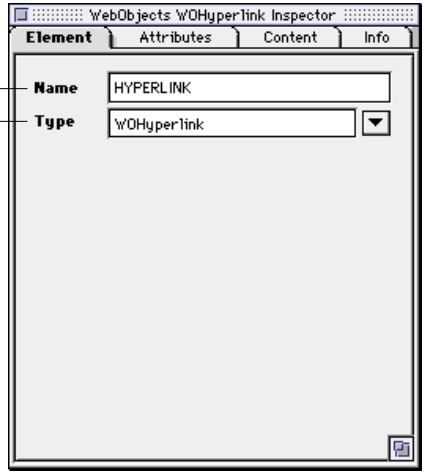
Inserting a WOHyperlink Tag



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

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

The Element Tab of the WOHyperlink Inspector



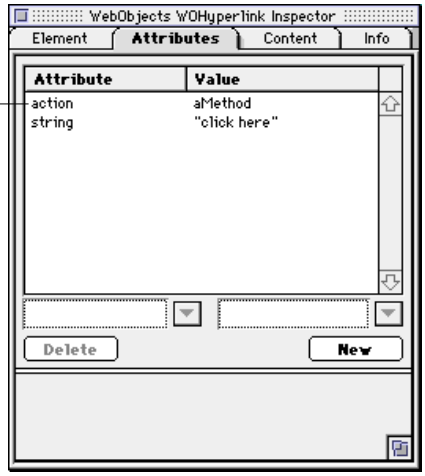
Set the following options in the *Element* tab of the *WOHyperlink Inspector*:

- 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 WebObject, if required.

The Attributes Tab of the WOHyperlink Inspector

The list box shows the existing attributes.

Use these options to add and delete attributes.



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

To add and delete attributes, proceed as follows:

- 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 have **GoLive CyberStudio** 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.

• 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.

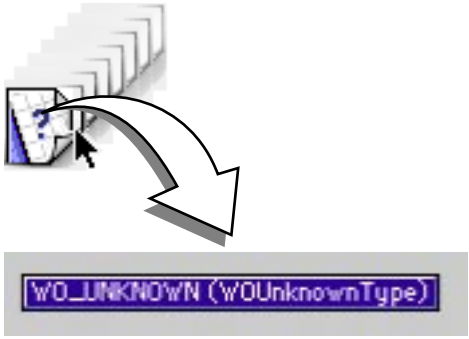
ReusableComponent

The *ReusableComponent* icon inserts an unknown dynamic element.

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

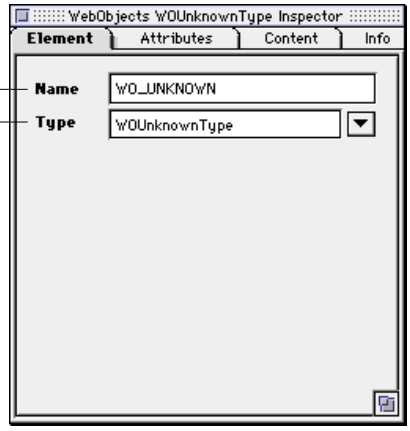


2. Select the *WOUnknownType* tag to set up the unknown element in the context-sensitive *Inspector* window, now titled *WOUnknown-Type Inspector.* The *WOUnknownType Inspector* has four tabs:
   - *Element* lets you set general properties for the current WebObject.
   - *Attributes* lets you edit attributes for the current WebObject.
   - *Content* displays object-specific text content, if any.
   - *Info* sbriefly describes the WebObjects element.
3. Control-doubleclick the icon in the document window to open the reusable component for editing.

The Element Tab of the WOHyperlink Inspector

Use this text box to name the WebObjects element.

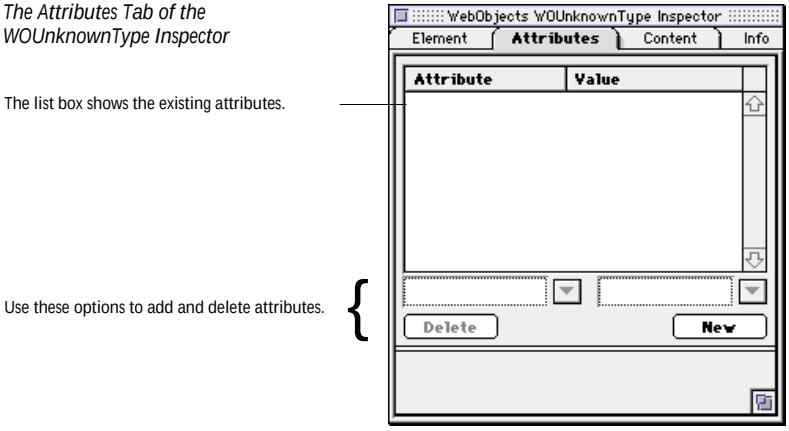
Use this popup menu as a shortcut to changing the type of the object.



Set the following options in the *Elementt*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 Database* on page 222).

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:

- Click the New button to add new attributes for the *WOUnknown-Type* 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 have **GoLive CyberStudio** 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.

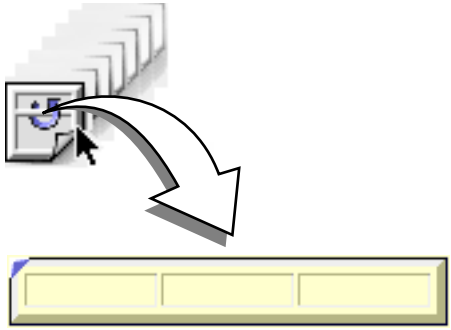
Table

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 the section *Tables* in the *GoLive CyberStudio User Manual*. For an update on new shortcuts and features, please see the section *Neue Kurzbefehle zur Tabellen-Bearbeitung*.