Using Dynamic Elements in WebObjects Builder

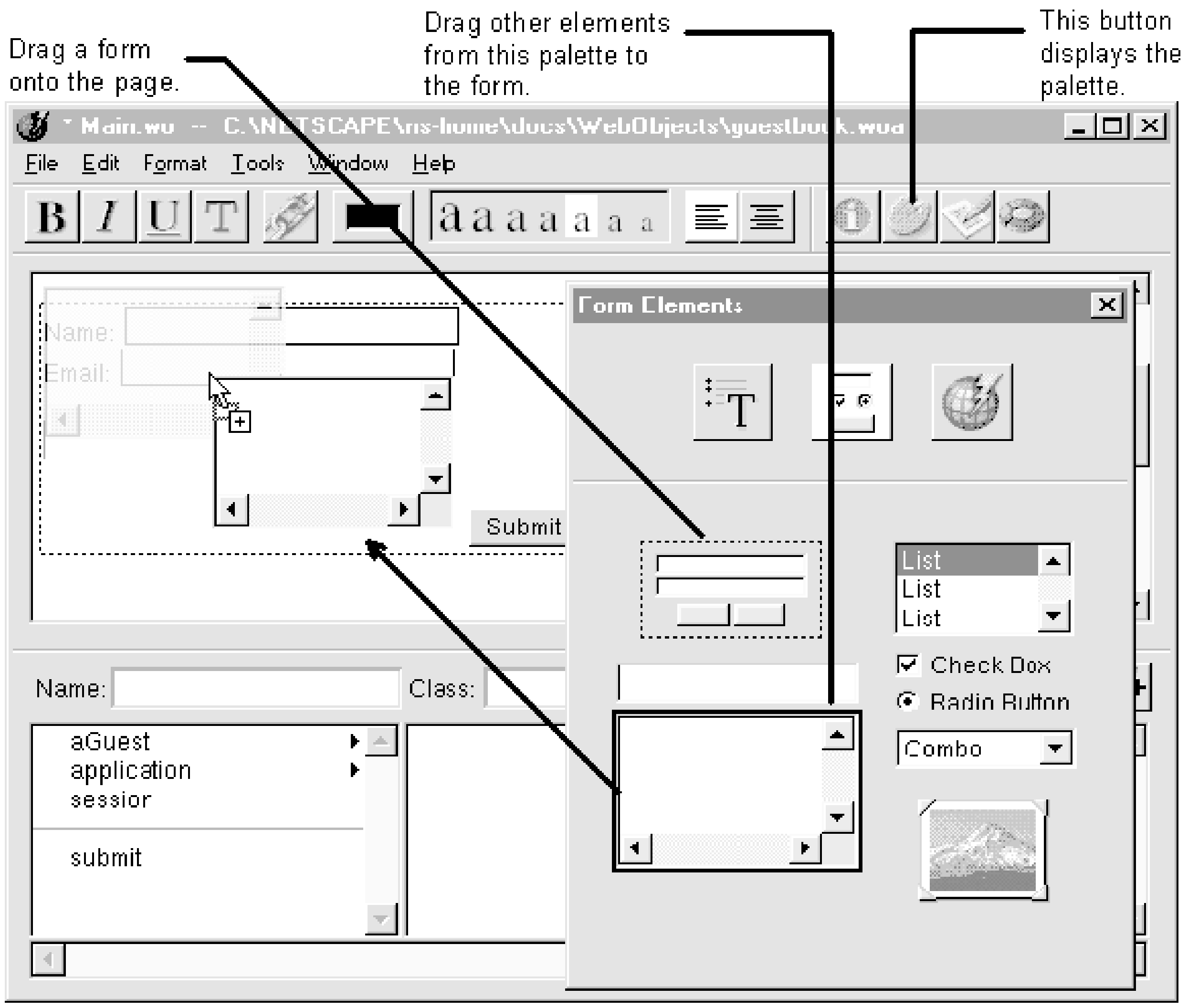
A dynamic element is an element that is replaced with dynamically generated HTML when the application runs. For example, WOString specifies a dynamic string element—the actual string it displays is determined at run time.

For a dynamic element to be fully functional, you must create the element and then bind it to some value. Usually, this value is a variable in a script file. If you bind the element to a variable, you must then write a script that sets the value of the variable (see "Writing WebScript in WebObjects Builder").

For more information on dynamic elements, see the chapter "How WebObjects Works" in the *WebObjects Developer's Guide*. Or look up individual dynamic elements in the "Dynamic Elements" section of the *WebObjects Reference* to learn more about them.

Creating Form-Based Dynamic Elements

1. Place the cursor where you want the form to appear on the page.
2. Drag the form element from the Form Elements palette onto the page.
3. Place the cursor inside the form.
4. Drag other elements from the Form Elements palette into the form.


Drag a form onto the page.
Drag other elements from this palette to the form.
This button displays the palette.
Main.wo -- C:\NETSCAPE\ns-home\docs\WebObjects\guestbook.woa
File Edit Format Tools Window Help
Form Elements
Name:
Email:
Submit
List
Check Box
Radio Button
Combo
Name:
Class:
aGuest
application
session
submit

The easiest way to create a form is to start with the pre-made form on the Form Elements palette. This element creates a form with two text fields and Submit and Reset buttons. After placing the form element, you can modify it by adding or removing elements.

When you use the pre-made form, WebObjects Builder inserts a <FORM> tag before the first text field and a </FORM> tag after the Reset button, specifying a single form. If you add form elements without using the pre-made form (for example, if you add a text field to an empty page) or if you add form elements outside of the form, WebObjects Builder assumes you want to create a new form and places <FORM> tags before and after the element.

The Form Elements palette can generate either HTML form elements or their corresponding WebObjects dynamic elements (WOTextField, WOSubmitButton, and so on). To generate the WebObjects dynamic element, all you have to do is create a binding for the element. If you don't create a binding, WebObjects Builder generates an HTML form element.

For example, if you drag a text field to a component and then immediately save the component, the text field is an HTML text field:

```
<FORM><INPUT type=text></FORM>
```

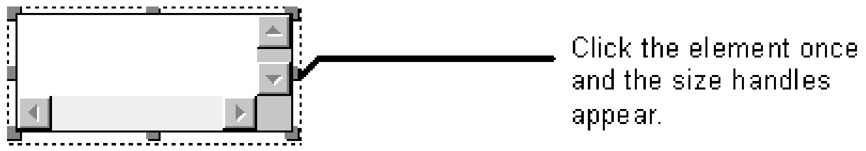
If you bind the text field to a value and save the component again, WebObjects Builder converts the HTML text field into a WebObjects WOTextField element:

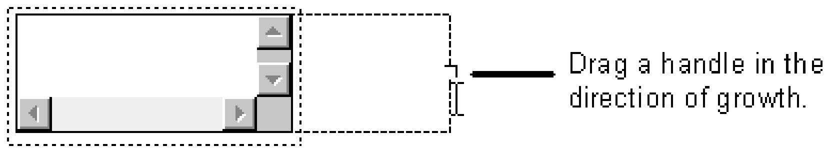
```
<FORM><WEBOBJECT name=TextField1><INPUT type=text value = guest></WEBOBJECT></FORM>
```

Tip: HTML forms don't allow you to have multiple submit buttons in a single form, but the WebObjects WOForm element does. If you want multiple submit buttons in a form, bind the **multiplesubmit** attribute of WOForm to the value 1. See "Binding Elements Using the Inspector" to learn how to do this. (To create the second submit button, use the menu command Format->Form->Add Submit Button.)

Resizing Elements

1. Click the element once.
2. Drag a size grip in the direction you want.

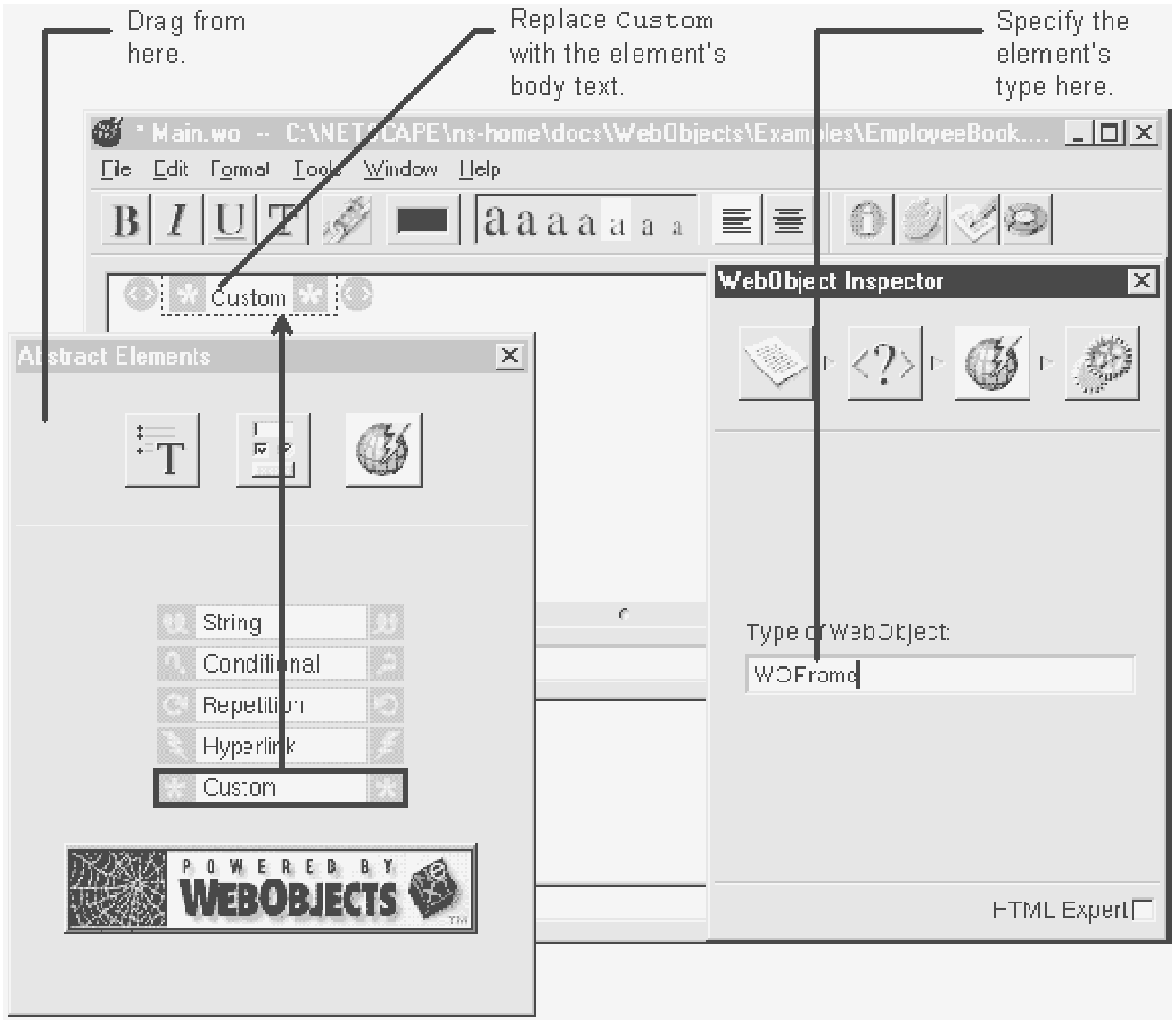




See "Formatting Tips" for other HTML formatting tips.

Creating Abstract Elements

1. Place the cursor where you want the element.
2. Choose the Abstract Elements palette. (Click the palette button to bring up the palette window.)
3. Drag the item from the palette onto the page.
4. If necessary, specify the element type in the inspector window. (Click the inspector button to bring up the inspector window.)


Drag from here.
Replace Custom with the element's body text.
Specify the element's type here.
* Main.wo -- C:\NETSCAPE\ns-home\docs\WebObjects\Examples\EmployeeBook....
File Edit Format Tools Window Help
Custom
Abstract Elements
String
Conditional
Repetition
Hyperlink
Custom
POWERED BY WEBOBJECTS
WebObject Inspector
Type of WebObject:
WOForm
HTML Expert

Abstract elements are dynamic elements that don't have an HTML counterpart. You can learn more about abstract elements and other dynamic elements in the chapter "How WebObjects Works" of the *WebObjects Developer's Guide*.

WOConditionals, WOHyperlinks, WORepetitions, and WOStrings don't need any extra setup in the inspector. To create any other type of abstract element, use the custom element and specify the element type in the inspector window. Once you have the element set up, you may want to store it on a custom palette if you are going to use it frequently.

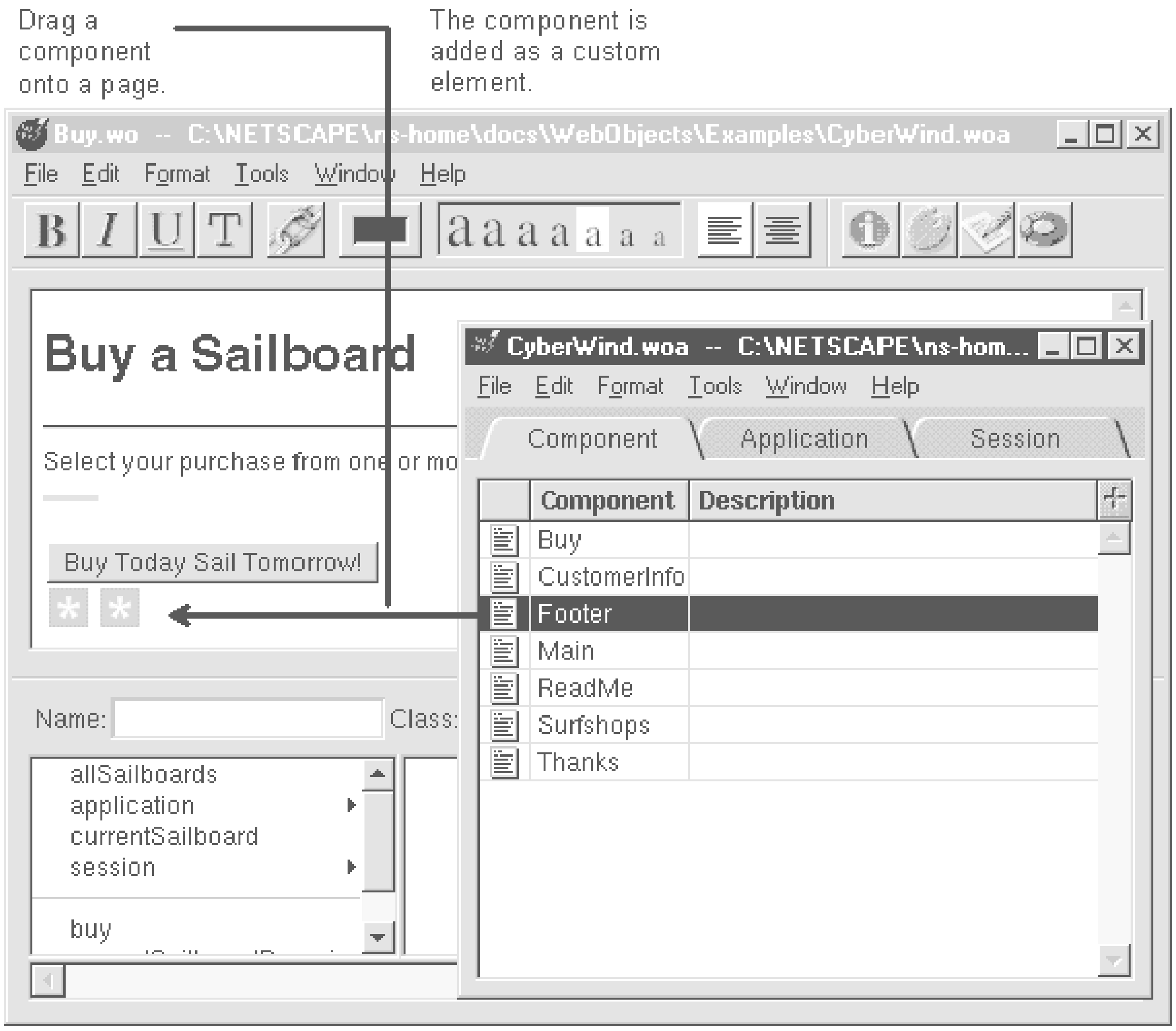
See "Formatting Tips" if you're having trouble editing the page.

The Custom Abstract Element

If the element you want to create isn't explicitly listed on the Abstract Elements palette, use the custom element. This creates an abstract dynamic element, but it leaves the type undefined. You enter the element's name in the inspector window. After you have done this, WebObjects Builder displays the binding attributes that element supports in the element's bindings inspector.

Reusing Components

1. Place the cursor where you want the component to appear.
2. Drag the component from an application window.


Drag a component onto a page.
The component is added as a custom element.
Buy.wo -- C:\NETSCAPE\ns-home\docs\WebObjects\Examples\CyberWind.woa
File Edit Format Tools Window Help
Buy a Sailboard
Select your purchase from one or mo
Buy Today Sail Tomorrow!
Name:
Class:
allSailboards
application
currentSailboard
session
buy
CyberWind.woa -- C:\NETSCAPE\ns-hom...
File Edit Format Tools Window Help
Component
Application
Session
Component
Description
Buy
CustomerInfo
Footer
Main
ReadMe
Surfshops
Thanks

Dragging a component into the component window adds it to the page as an abstract element. The component looks like a custom element. If you inspect this element, you'll see it has the component's name. You treat the component just like you would any other dynamic element: you must bind to it to be able to interact with it in the script file.

You can drag a component from any application window. If you drag across applications, the component is added to the destination component's application (because WebObjects requires that all components used by an application be in that application's directory.)

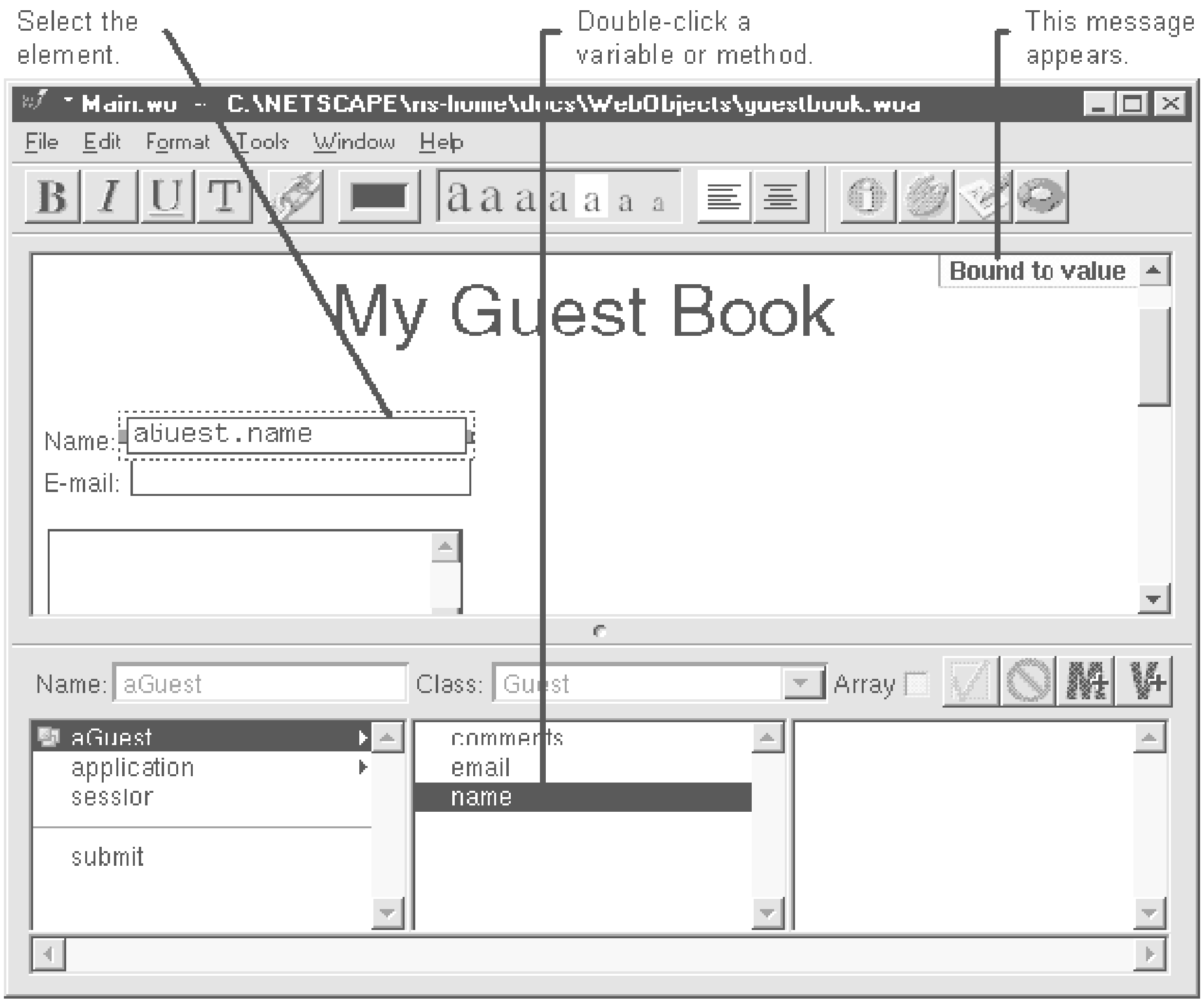
To create components that can be reused in this manner, see "Creating Reusable Components."

Tip: If you use a component frequently, store it on a custom palette.

Binding Elements

1. Select an element.
2. Double-click a variable in the object browser.

A message appears in the upper right corner of the component window's editing display. The message tells you what attribute the variable is bound to.



Bindings are the key to writing dynamic HTML pages in WebObjects. A *binding* is a mapping between a variable or method declared in a component's script and a dynamic element in the component's HTML.

Each dynamic element defines one or more *attributes*. You can bind each attribute to a different variable or method in your script. Thus, there is not a one-to-one correlation between dynamic elements and variables. To bind an element, you must specify three things: the variable (or method), the element, and the attribute within the element.

Many times, WebObjects Builder can determine which attribute you want to bind to without you having to explicitly specify the attribute. For example, if you select a WORepetition and double-click an array, WebObjects Builder binds the array to the list attribute. If you select a WORepetiton and double-click a variable that isn't an array, WebObjects Builder binds the variable to the item attribute (which represents a single item in the WORepetition's list).

For each dynamic element, WebObjects Builder chooses the attribute that is most commonly used, given the class of the double-clicked variable. Thus, double-clicking a variable should produce the binding you want most of the time.

You can override WebObjects Builder's default bindings by binding using the inspector instead of double-clicking. You might need to do this when:

- You want to bind to a constant value.

Most dynamic element attributes can be bound to constants as well as variables. If you want to bind to a constant, you must use the inspector interface.

- WebObjects Builder cannot determine which attribute to bind to.

  This happens when the dynamic element doesn't have one attribute that's more commonly used than its other attributes. If WebObjects Builder can't determine a default attribute, it displays the message "No suitable default binding" and you can complete the binding using the inspector.

- You double-clicked a variable whose class doesn't match the class accepted by the dynamic element's default attributes.

  For example, WOString's only default attribute is **value**, which specifies what the string should display. WebObjects Builder automatically binds to this attribute only if you double-click a simple variable (that is, an Object, Number, or String, all of which can easily translate into a string). If you double-click an array, WebObjects Builder displays the message "No suitable default binding." If you truly intend to bind the WOString to the array, you can do so using the inspector.

- You want to bind to an attribute that is not the default.

  For example, suppose you want to bind to the WOString's **escapeHTML** attribute (which determines if any HTML tags in the string should be interpreted by the browser or displayed as specified). If you select the WOString and double-click a variable, WebObjects Builder chooses the **value** attribute.

- You cannot easily select the element you want to bind.

  For example, if you have a WORepetition that surrounds a table row, the WORepetition doesn't appear in the component window.

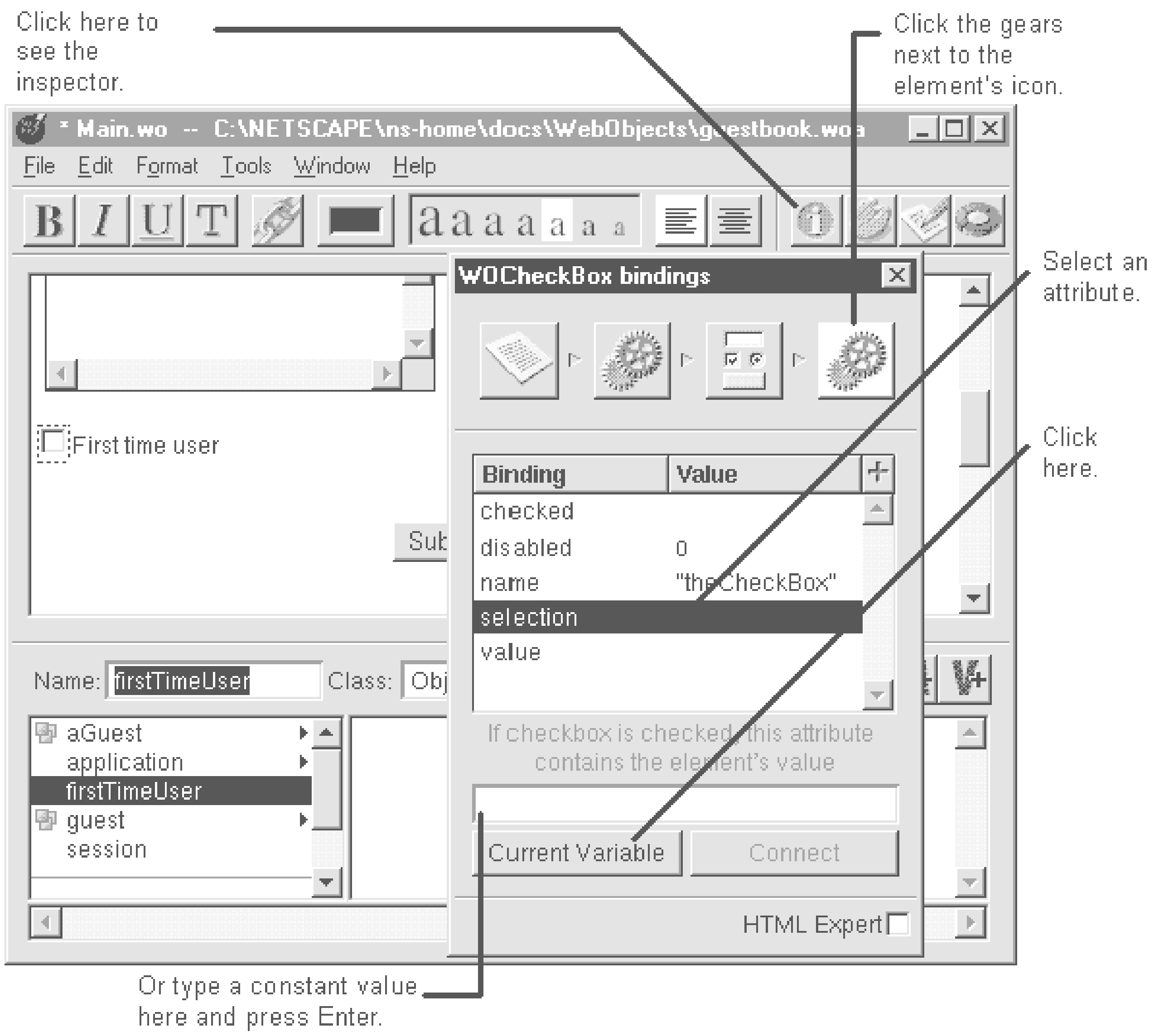
To learn more about a dynamic element's attributes, look it up in the "Dynamic Elements" section of the *WebObjects Reference*. Also, you can select the attribute in the bindings inspector to see a short description of the attribute.

Binding Elements Using the Inspector

1. Select the element.
2. Select a variable or method in the object browser.
3. In the bindings inspector, select the attribute you want to bind to. (Click the inspector button to display the inspector window.)
4. Click the Current Variable button.

OR:

1. Select the element.
2. In the bindings inspector, select the attribute you want to bind to.
3. Type a value in the inspector's text field. **Note**: If the value is a string constant, you must put it in quotation marks.
4. Press Connect.


Click here to see the inspector.
Click the gears next to the element's icon.
Select an attribute.
Click here.
Or type a constant value here and press Enter.
* Main.wo -- C:\NETSCAPE\ns-home\docs\WebObjects\guestbook.woa
File Edit Format Tools Window Help
WOCheckBox bindings
First time user
Name: firstTimeUser
Class: Obj
aGuest
application
firstTimeUser
guest
session
Current Variable
Connect
HTML Expert

Binding	Value
checked	
disabled	0
name	"theCheckBox"
selection	
value	

If checkbox is checked, this attribute contains the element's value

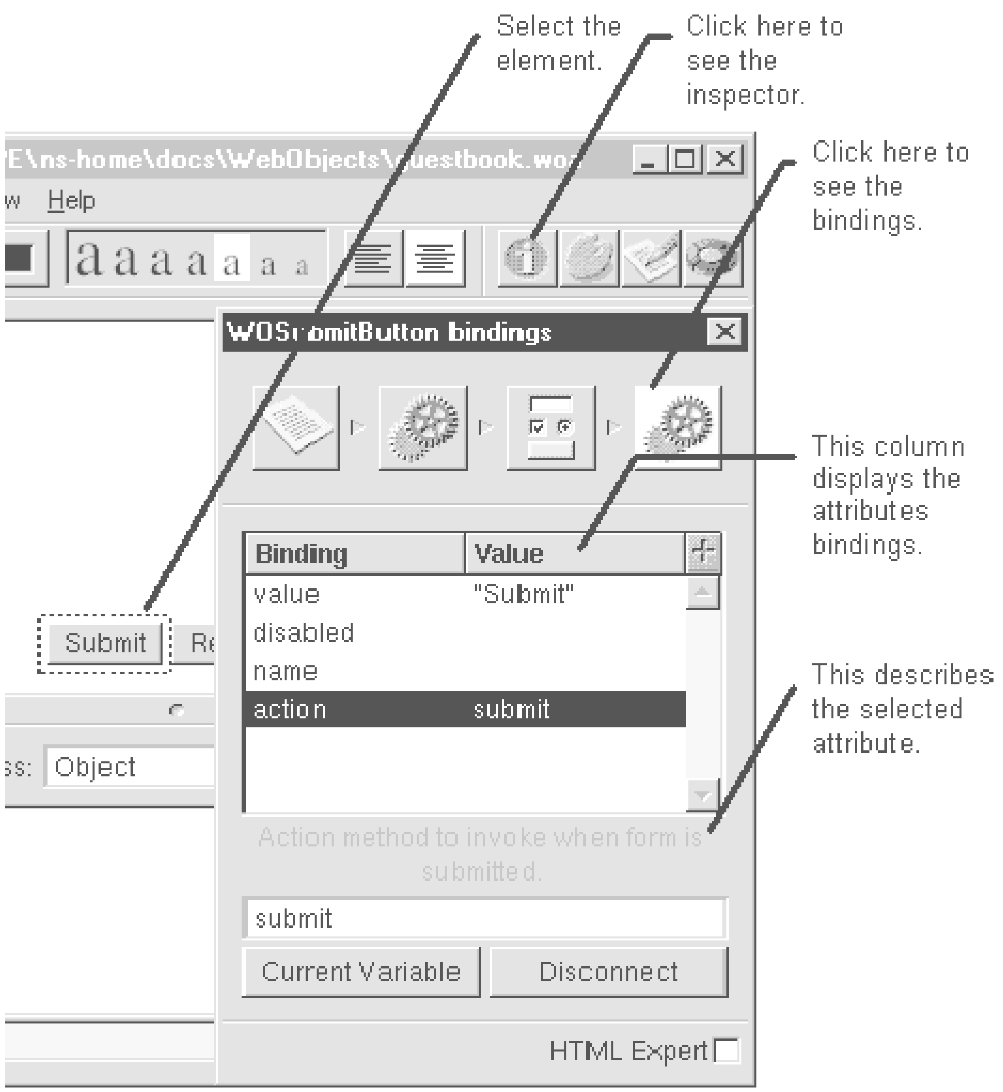
Usually, you don't need to use the inspector window to bind elements. Instead you can just select the element and double-click a variable. See "Binding Elements" for more information on how this works and for a list of cases where you might need to bind using the inspector.

Sometimes, it's difficult to select the element you want to bind. For example, if you have a WORepetition that surrounds a table row, the WORepetition doesn't appear in the component window, and you can't select it. In this case, you can select the table row and then use the inspector's icon path to navigate to the WORepetition's bindings inspector. See "Selecting Elements in WebObjects Builder" for more information.

Displaying Bindings

1. Select the element in the component window.

2. Click the element’s bindings inspector icon. (Click the inspector button to bring up the inspector window.)


Select the element.
Click here to see the inspector.
Click here to see the bindings.
This column displays the attributes bindings.
This describes the selected attribute.
WOSubmitButton bindings
Help
Submit
Current Variable
Disconnect
HTML Expert

Binding	Value
value	"Submit"
disabled	
name	
action	submit

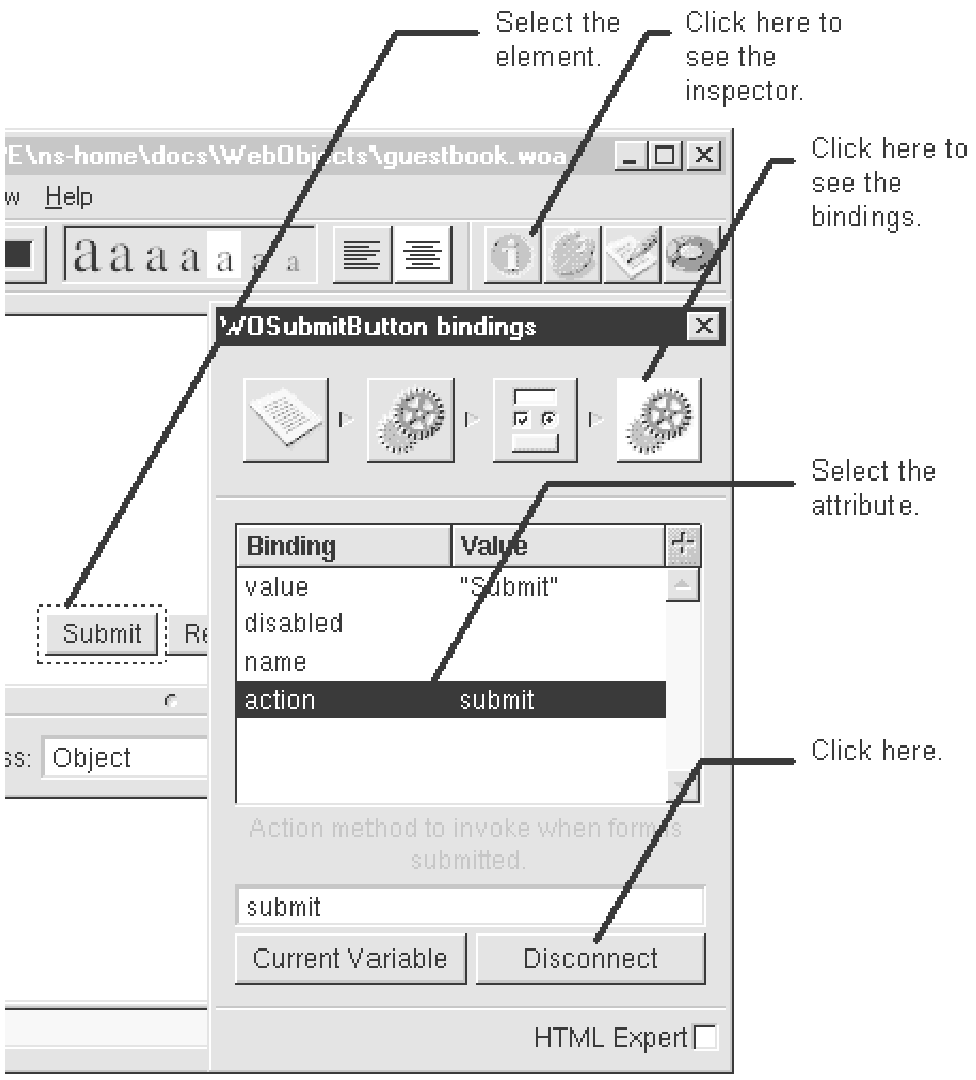
Action method to invoke when form is submitted.

submit

Undoing a Binding

1. Select the element in the component window.
2. Click the element’s bindings inspector icon. (Click the inspector button to bring up the inspector window.)
3. Select the attribute.
4. Click the Disconnect button.

Note: You can change an attribute’s binding without having to disconnect.



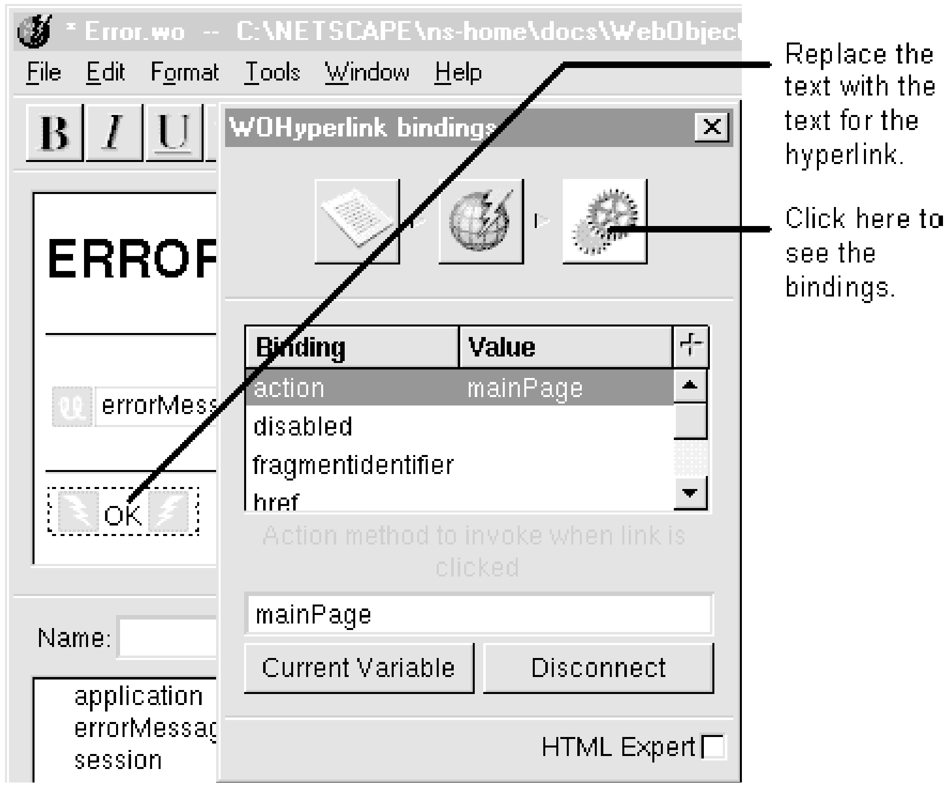
Binding	Value
value	"Submit"
disabled	
name	
action	submit

Action method to invoke when form is submitted.

submit

Creating Dynamic Hyperlinks

1. Drag a WOHyperlink from the Abstract Elements palette to the component window.
2. Replace the word `Hyperlink` with the text of the link.
3. Bind the hyperlink to a variable or a method.


* Error.wo -- C:\NETSCAPE\ns-home\docs\WebObject
File Edit Format Tools Window Help
B I U
ERROR
errorMess
OK
Name:
application
errorMessag
session
WOHyperlink bindings
Action method to invoke when link is clicked
Current Variable
Disconnect
HTML Expert
Replace the text with the text for the hyperlink.
Click here to see the bindings.

Binding	Value
action	mainPage
disabled	
fragmentidentifier	
href	

mainPage

WebObjects Builder allows you to create either static hyperlinks or dynamic hyperlinks (which are WOHyperlink dynamic elements). Use WOHyperlink instead of a static hyperlink when:

- You want to link to a page that's returned by a component in your application. Pages in a WebObjects application don't have predictable URLs that you can specify in an HTML hyperlink.

  In this case, you can bind to WOHyperlink's pageName attribute. Specify the component's name in the bindings inspector's text field. See "Binding Elements Using the Inspector."

- You want the hyperlink to perform an action rather than return a page.

  In this case, you can bind to WOHyperlink's action attribute. Specify the method that should be invoked when the link is clicked.

- You want the hyperlink's destination to be determined at run time.

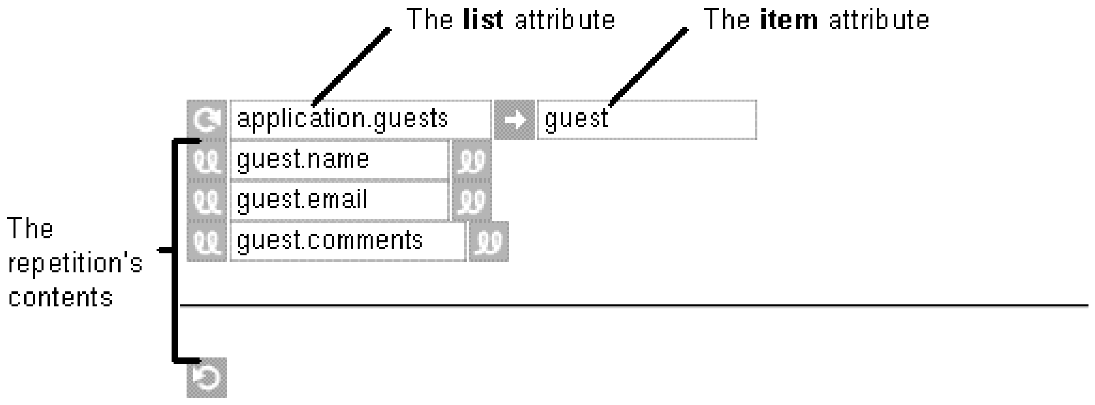
  In this case, you can bind to WOHyperlink's action attribute. Specify a method that determines which page should be returned when the link is clicked.

The inspector shows a complete list of the WOHyperlink attributes you might bind to, and the WOHyperlink description in the *WebObjects Reference* contains complete descriptions of them.

To learn how to create a static hyperlink, see "Creating Hyperlinks."

Binding to WORepetitions

1. Select the WORepetition.
2. Double-click an array variable in the object browser.



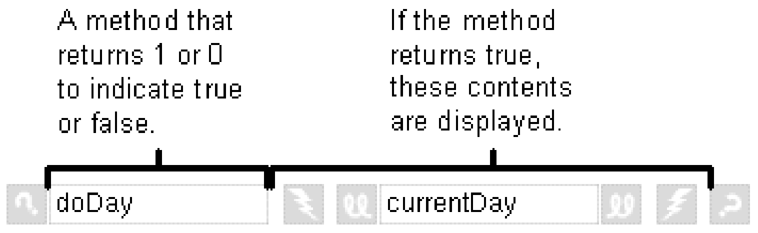
WORepetition has two attributes that you must bind to: **list** and **item**. The **list** attribute must be bound to an array. The **item** attribute is bound automatically when you bind the **list** attribute.

A WORepetition is like a loop in a structured programming language. It uses **item** to iterate through the **list**. Creating a WORepetition is equivalent to saying "for each **item** in the array **list**, display the contents."

Tip: It's common to use WORepetitions with tables. To learn how to bind a WORepetition that has a table row as its contents, see "Binding Elements Using the Inspector."

Binding to WOConditionals

1. Select the WOConditional.
2. Double-click a variable in the variables browser.


A method that returns 1 or 0 to indicate true or false.
If the method returns true, these contents are displayed.
doDay
currentDay

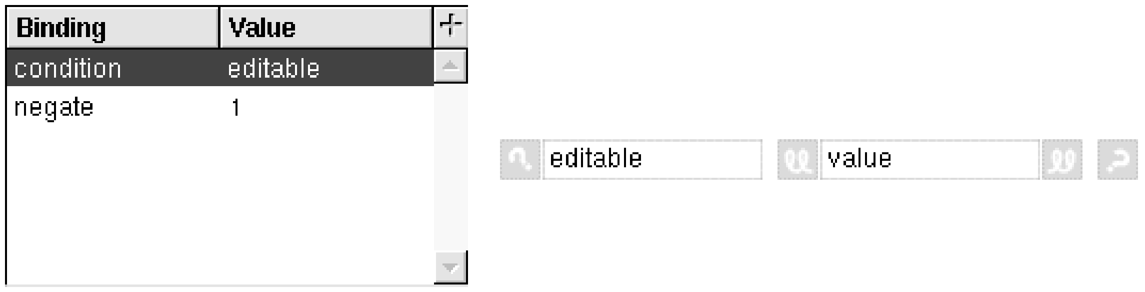
A WOConditional displays its contents only if a particular condition is true. WOConditional's main attribute is **condition**. If **condition** is 1 (true), the WOConditional's contents are displayed. If **condition** is 0, the contents aren't displayed. **condition** can be bound to a variable or to method that returns the 1 or 0 value. The values self or nil also work in place of 1 or 0.

A WOConditional is like an if-then statement in a structured programming language. It tests to see if its condition is true, and if so, it displays its contents. If the condition is false, it displays nothing.

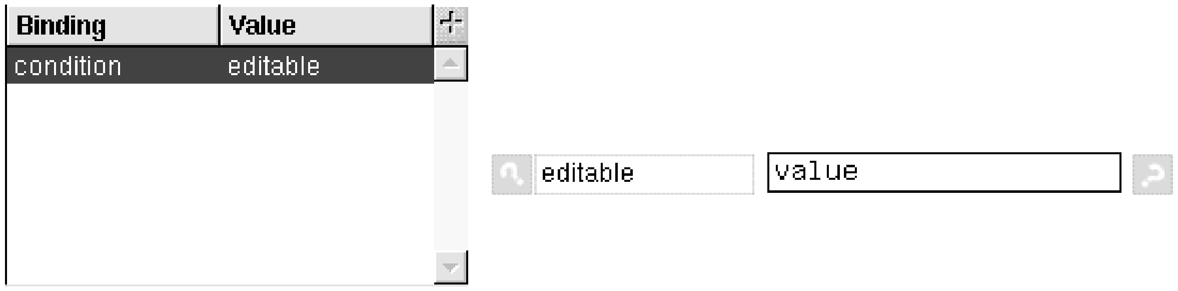
Many times, you want an “else” clause as well; that is, “if the condition is true, display this text; if not, display this other text.” In such cases, you can make use of another WOConditional attribute: negate. If negate is 1, it means that the condition is negated after it is evaluated. Thus, if negate is 1 and condition returns 1, the contents are *not* displayed. To create an if-then-else style of conditional contents, do the following:

1. Drag two WOConditionals onto the page.
2. Select the first conditional and double-click a variable in the object browser.
3. Select the second conditional and double-click the same variable.
4. In the bindings inspector for the second WOConditional, select the negate attribute, type 1 in the text field in the bindings inspector, and click Connect.

The example below is from a component named ReadWriteString. This component lets you set whether a displayed string is editable using the variable editable. If editable is 1, the component displays the WOTextField contained in the second conditional. The WOString from the first conditional is not displayed because the first conditional negates the value in editable. If editable is 0, the WOString is displayed but not the WOTextField.



Binding	Value
condition	editable
negate	1



Binding	Value
condition	editable