Setting Up Applications in WebObjects Builder

A WebObjects application consists mainly of components. You create an application using WebObjects Builder, then you create or add the components you want it to have. Applications also may have an application script and a session script. The sections listed here tell you how to use WebObjects Builder to create an application.

For more information on what a WebObjects application is and what it contains, see "The Ingredients of a WebObjects Application" in the *WebObjects Developer's Guide*.

Creating Applications

Note: To create an application that has compiled code, use Project Builder instead of WebObjects Builder. See "Compiling and Debugging a WebObjects Application" in the *WebObjects Developer's Guide*.

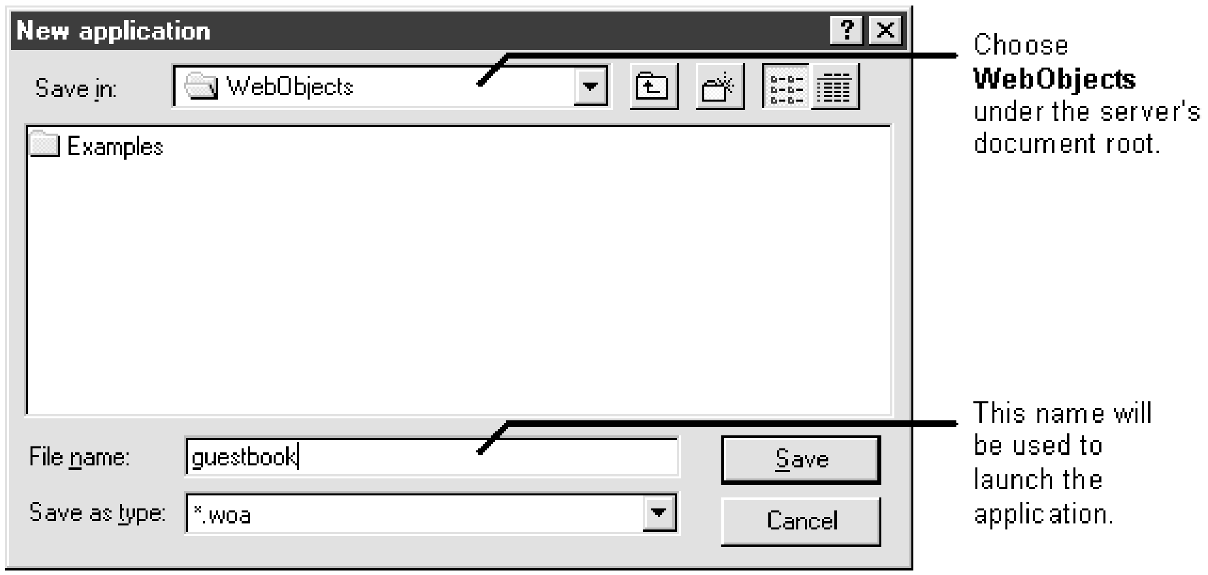
To create a new application in WebObjects Builder:

1. Choose File->New Application.

   The panel that opens shows you the contents of <*DocumentRoot*>/WebObjects (unless you have previously saved into a different directory). <*DocumentRoot*> is your HTTP server's document root, which you specified when you installed WebObjects.

2. Enter a name for the application.

3. Click Save to create the application directory.


New application
Save in:
WebObjects
Examples
Save
Cancel
Choose **WebObjects** under the server's document root.
This name will be used to launch the application.

File name: guestbook

Save as type: *.woa

WebObjects Builder creates the application directory *AppName*.woa, where *AppName* is the name you specified. Under *AppName*.woa, WebObjects Builder creates another directory called Main.wo for your first page, or *component*. (See "Layout of Application in the File System.")

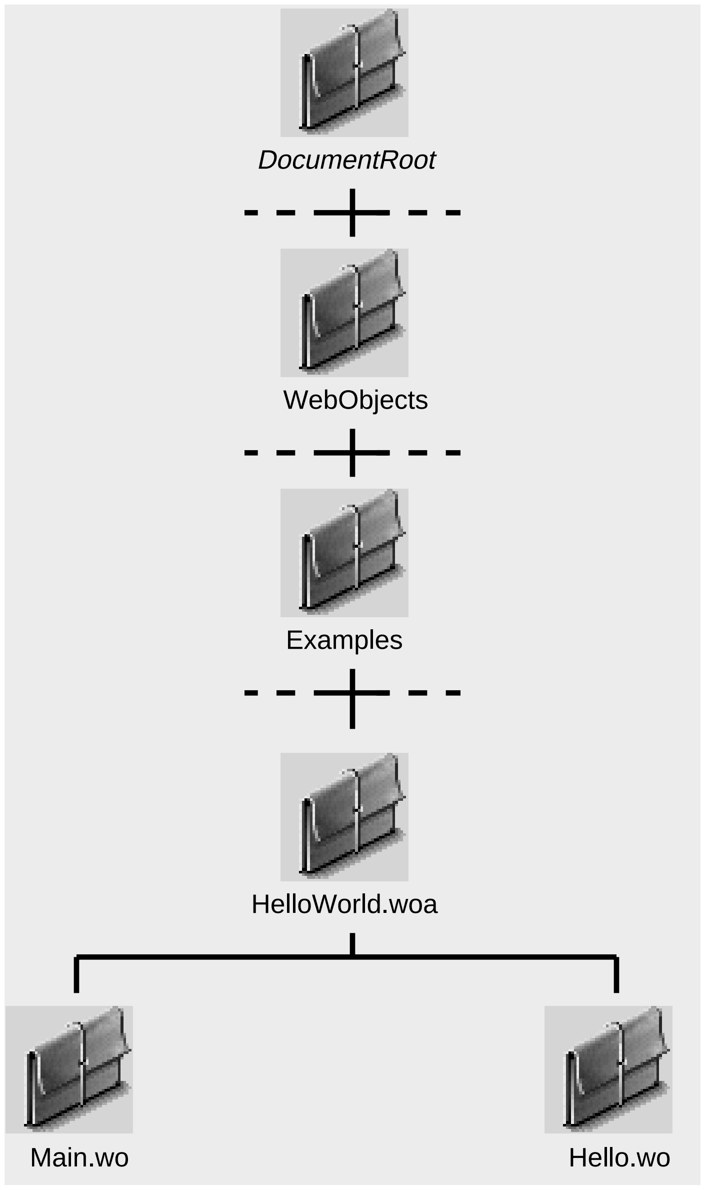
An empty editing window for Main.wo opens along with an application window that lists all the components in your application. At first only Main.wo is listed in the application window.

Layout of Application in the File System

When you create an application in WebObjects Builder, you're creating a new directory under <DocumentRoot>/WebObjects. All WebObjects applications must be located beneath this directory so that the HTTP server can find them. The application directory is given the extension .woa.

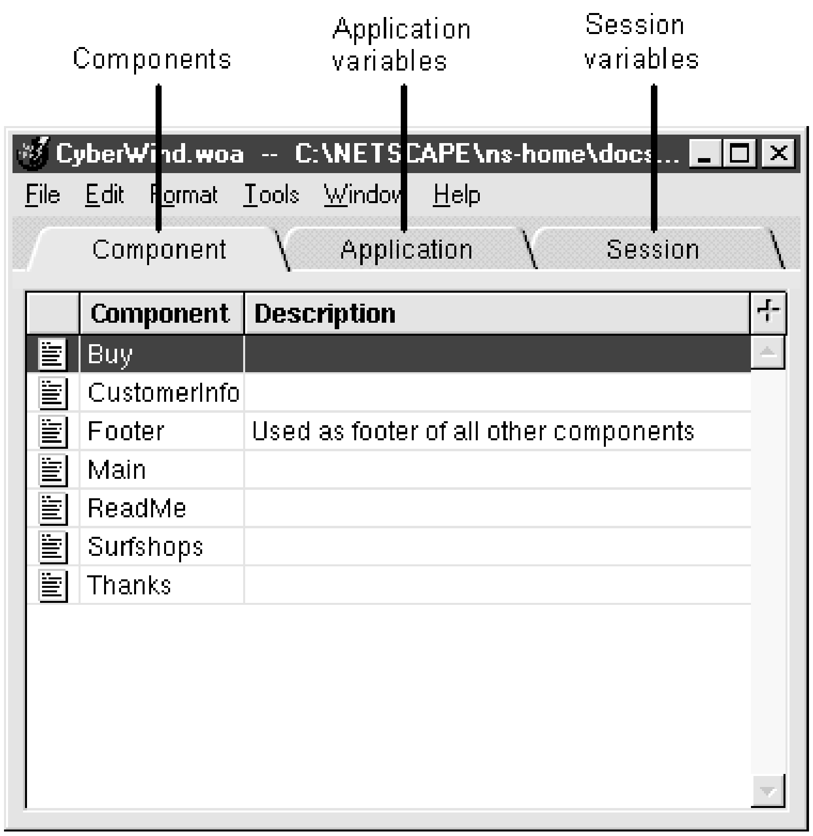
The application directory contains the application's resources. These resources consist mainly of components, which are dynamic HTML pages.

Components are themselves directories, consisting of files that store the HTML, the script, bindings, and any component-specific resources. The component for the first page of the application is usually named Main.wo.



Application Window in WebObjects Builder

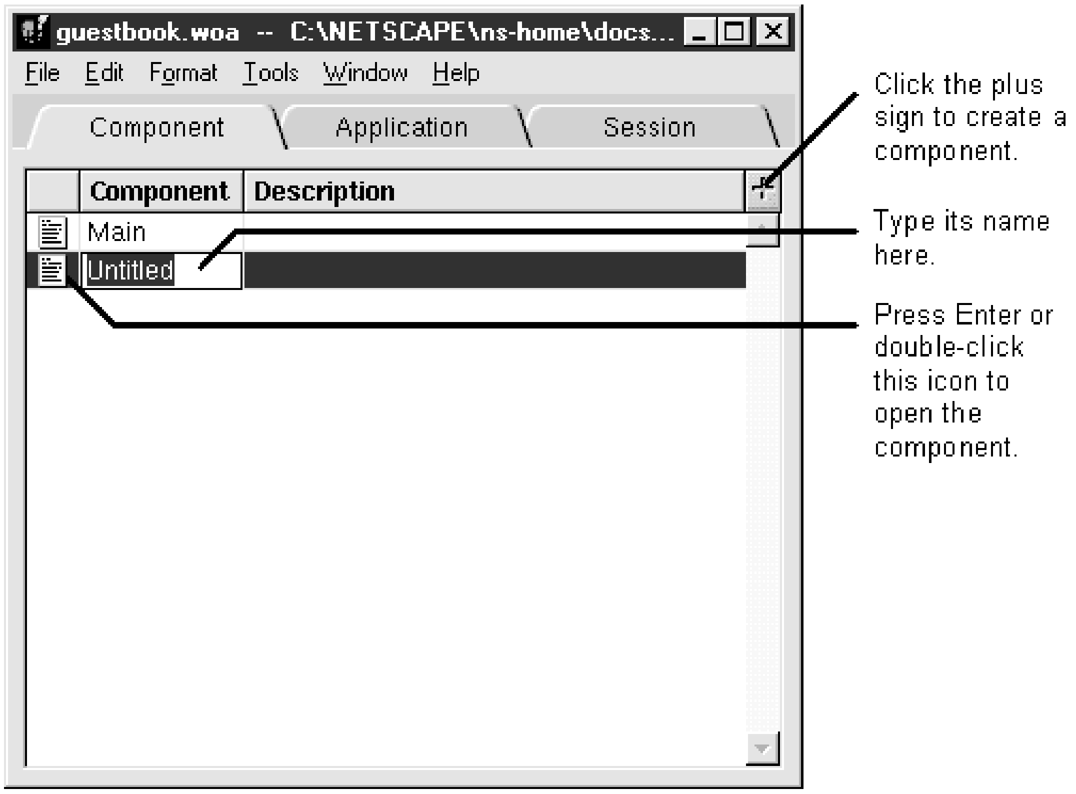
Use the application window to control application-wide resources. You use this window mainly to open, create, or delete components inside the application. The application window is also where you define application variables and session variables. Application variables are created when the application starts executing and are deallocated when the application quits. Session variables are created when a new user begins a new session and are deallocated when the session ends. Both application and session variables can be accessed by all of the application's components. "Writing WebScript in WebObjects Builder" describes how to use the application window to create application and session variables.


Components
Application variables
Session variables
CyberWind.woa -- C:\NETSCAPE\ns-home\docs...
File Edit Format Tools Window Help
Component
Application
Session

Component	Description
Buy	
CustomerInfo	
Footer	Used as footer of all other components
Main	
ReadMe	
Surfshops	
Thanks	

Creating Components

1. Click the plus sign in the Component display of the application window. A new component is created with the name `Untitled`.
2. Type a name for the component.
3. Press Enter.


guestbook.woa -- C:\NETSCAPE\ns-home\docs...
File Edit Format Tools Window Help
Component
Application
Session
Component
Description
Main
Untitled
Click the plus sign to create a component.
Type its name here.
Press Enter or double-click this icon to open the component.

If you have a large number of components, you might want to enter a comment in the Description field to help you identify the component.

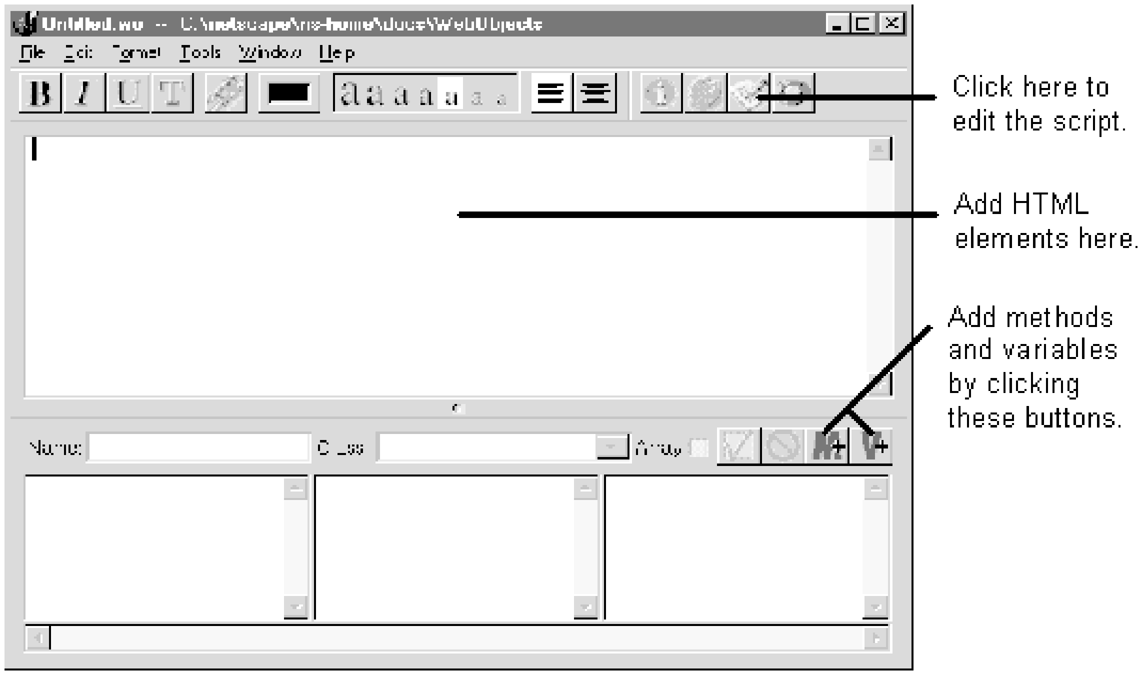
When you create a component, you're creating a subdirectory named *Component*.wo under the application directory. There are three parts to a component: an HTML template, a script, and bindings between the two. You use WebObjects Builder's component window to create these three parts. "Components" in the introduction to the *WebObjects Developer's Guide* describes other things that can go into a component.

In many cases, you'll want to use components that already exist. Reusing components is one of the powerful features of WebObjects. If the component already exists, you can add it to the application.

Note: The menu command File->New creates a new component that is not added to your application.

Component Window

Use the component window to edit a component. There are three basic things you add to a component: HTML elements, variables, and methods. You add HTML elements in the upper display of the component window by dragging them from a palette. You add variables and methods to the bottom of the window by clicking the add variable and add method buttons in the bottom half of the window. Finally, you edit methods using the Script window, which you can see if you click the script button.

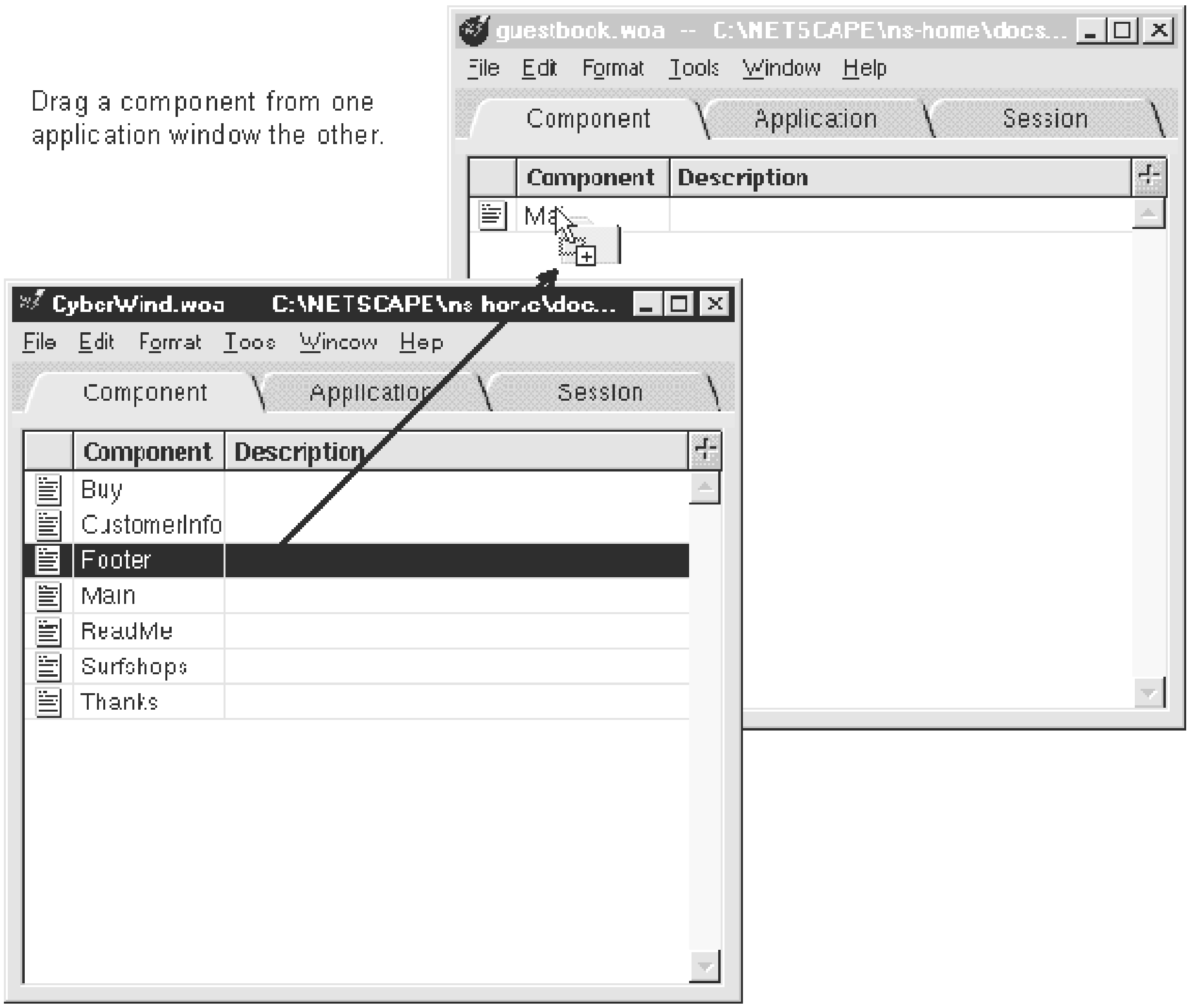

Untitled.wo -- C:\netscape\ns-home\docs\WebObjects
File Edit Format Tools Window Help
Name:
Class
Array
Click here to edit the script.
Add HTML elements here.
Add methods and variables by clicking these buttons.

“HTML Editing in WebObjects Builder” and “Using Dynamic Elements in WebObjects Builder” describe how to add HTML elements.

“Writing WebScript in WebObjects Builder” describes how to add variables and methods and edit the script.

Adding Existing Components

1. Open the application containing the component you want to reuse.
2. Drag the component from that application to the current application’s window.


Drag a component from one application window the other.
guestbook.woa -- C:\NETSCAPE\ns-home\docs...
File Edit Format Tools Window Help
Component Application Session
Component Description
Ma
CyberWind.woa C:\NETSCAPE\ns home\doc...
Component
Buy
CustomerInfo
Footer
Main
ReadMe
Surfshops
Thanks

You can reuse components that you created for other applications by dragging them into a page in the current application.

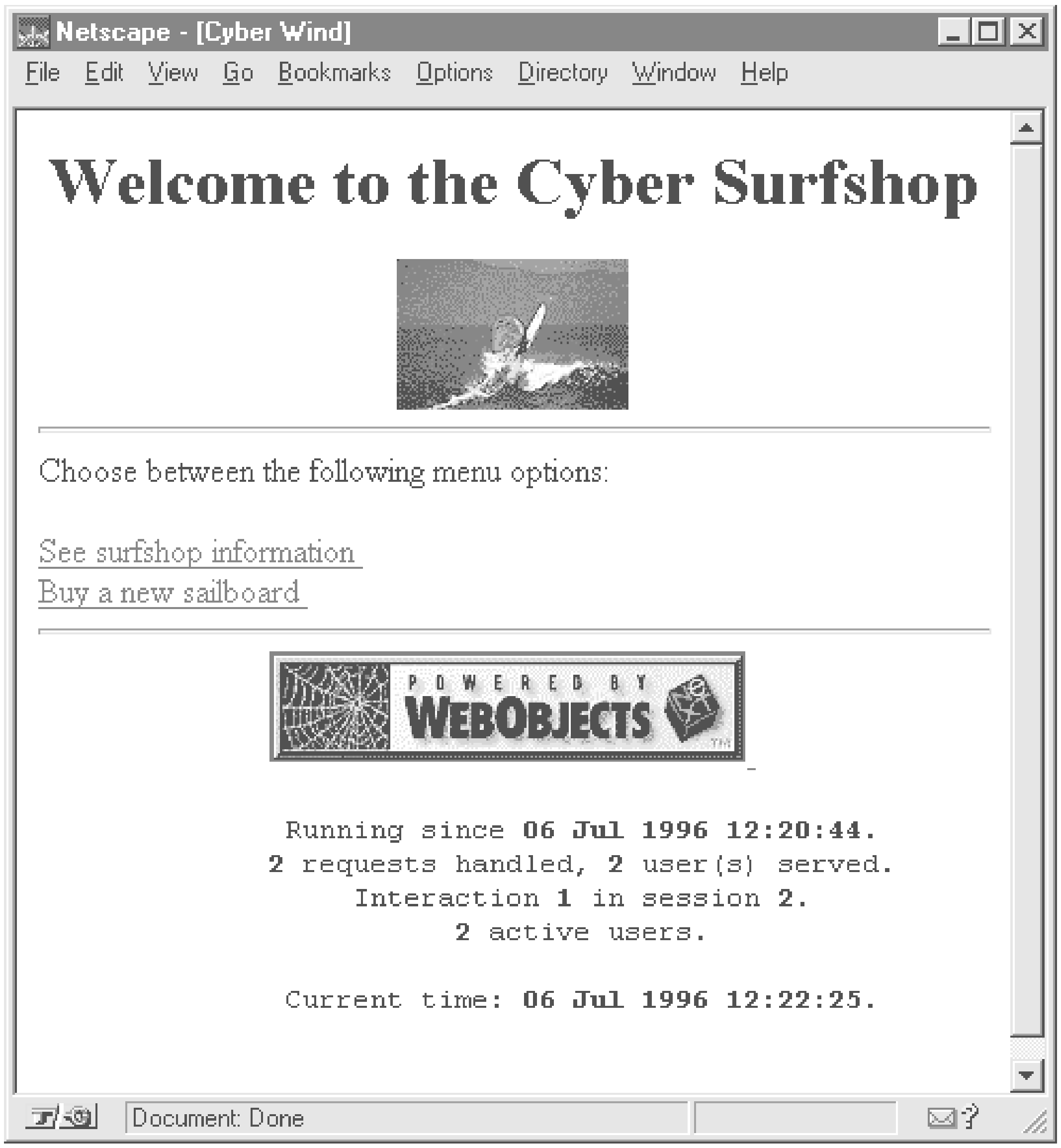
When you drag a component from one application window to another, the component is copied into the destination application's directory. Applications can only use components that are copied into the application directory.

If the existing component is intended to be used within a page, you can drag the component directly into a page in the destination application. If you do this, it adds the component to the page as a "custom element" (in addition to listing it in the application window). If you inspect this custom element, you'll see it has the component's name.

Reusable Components

One of the strengths of the WebObjects architecture is its support of reusable components. Any component that you define can be reused by any WebObjects application. The most common type of component represents an entire HTML page. However, WebObjects also supports components that represent a part of a page and can be used within multiple pages of the same application or even multiple sections of the same page.

Although some pages must be crafted individually for an application, many could be identical across applications. Even pages that aren't identical across applications can share at least some portions (header, footer, navigation bars, and so on) with pages in other applications. With reusable components, you can factor out a portion of a page (or a complete page) that's used throughout one or more applications, define it once, and then use it wherever you want, simply by referring to it by name.



Welcome to the Cyber Surfshop

Choose between the following menu options:

See surfshop information
Buy a new sailboard

```
Running since 06 Jul 1996 12:20:44.
2 requests handled, 2 user(s) served.
Interaction 1 in session 2.
2 active users.

Current time: 06 Jul 1996 12:22:25.
```

The footer is a component that's reused on all pages.

To reuse an existing component, you must add it to each application that wants to use it.

When you add one component to the page of another component, WebObjects Builder displays it as a custom element. You use this component just like you would use any other dynamic element that you drag from the palette.

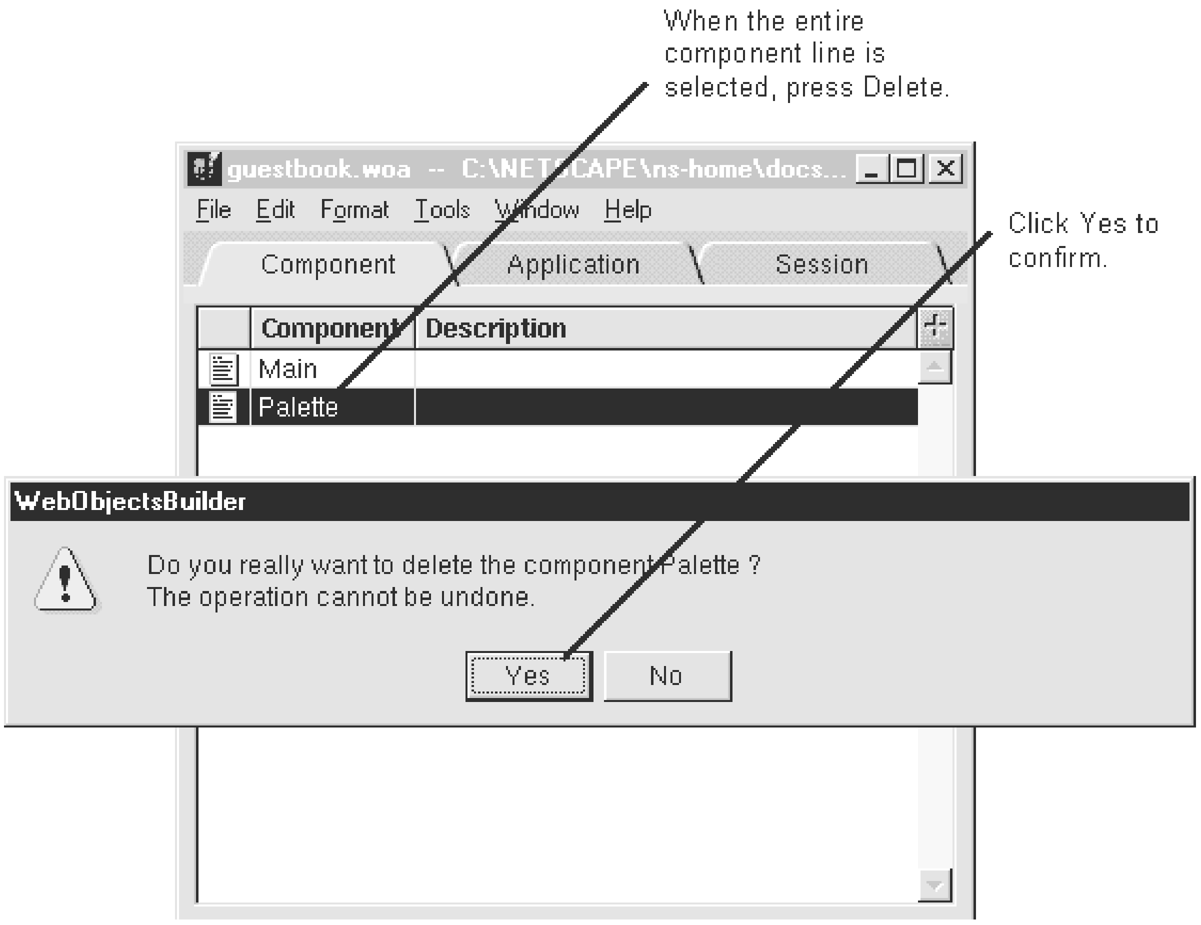
For more information on reusable components, see “Creating Reusable Components” in the *WebObjects Developer's Guide*. Also, “Reusing Components” and “Creating Reusable Components” in this guide describe how to use a component within another component and how to create a reusable component using WebObjects Builder.

Deleting Components

1. Click once to select the component in the application window.

2. Press the Delete key.

3. In the message box that opens, click the Yes button.


When the entire component line is selected, press Delete.
Click Yes to confirm.
guestbook.woa -- C:\NETSCAPE\ns-home\docs...
File Edit Format Tools Window Help
Component
Application
Session
Component
Description
Main
Palette
WebObjectsBuilder
Yes
No

Do you really want to delete the component Palette ?
The operation cannot be undone.

Sharing Resources

When you add a resource such as an image file to a component, the default behavior is to copy the file into the component's directory. The HTML that WebObjects Builder generates refers to the copy inside the component directory.

You may have resources that you want to share among several applications and several components. To set this up:

1. Create a directory under your HTTP server's document root.

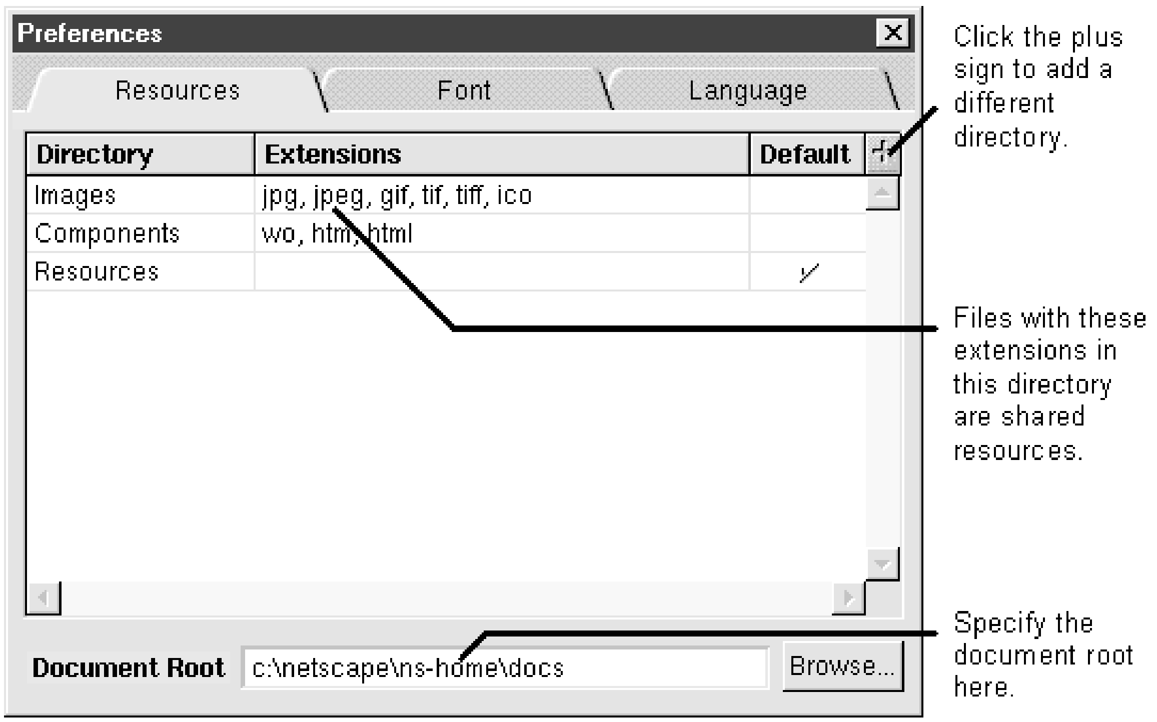
2. Specify that directory's name in WebObjects Builder's Preferences panel.

WebObjects Builder already assumes that you have two directories under the document root: Images and Resources. If they don't already exist, create these two directories. Use Images to store image files (.gif, .jpeg, .tiff, and so on). Use Resources to store all other files. You only need to add other directory names to the Preferences panel if you want to further organize your resources. For example, you might want to store all of the sound files in a separate directory named Sounds. In this case, you would create the Sounds directory and then add it to the table in the Preferences panel.

If you drag a resource from one of the directories specified on the Preferences panel, WebObjects Builder assumes you want to share the resource and does not copy it into the component. If the Preferences panel lists extensions next to the directory, it only shares files with that extension.

For example, if you drag a .gif file from the directory <*DocumentRoot*>/Images to a component, that file is not copied into the component directory because the Preferences panel specifies that all .gif files in <*DocumentRoot*>/Images are shared resources. However, if there was a sound file in <*DocumentRoot*>/Images and you dragged it to a component, the sound file *would* be copied into the component because the Preferences panel does *not* say that sound files in <*DocumentRoot*>/Images are shared resources.

If you don't specify file extensions next to the directory name on the Preferences panel, any file in that directory is shared. For example, all files in the Resources directory are shared, regardless of their type.



Directory	Extensions	Default
Images	jpg, jpeg, gif, tif, tiff, ico	
Components	wo, htm, html	
Resources		✓

Document Root c:\netscape\ns-home\docs

Click the plus sign to add a different directory.

Files with these extensions in this directory are shared resources.

Specify the document root here.

Note: All of an application's components must reside in the application's directory. You cannot create a shared component directory under the document root. See "Adding Existing Components" to learn how to share components.