Inside WebObjects

Deploying WebObjects Applications

August 2001

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Computer, Inc.
Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple-labeled or Apple-licensed computers

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Mac, Macintosh, and WebObjects are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Enterprise Objects and Enterprise Objects Framework are trademarks of NeXT Software, Inc., registered in the United States and other countries.

Java is a registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Simultaneously published in the United States and Canada

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Figures, Listings, and Tables

Chapter 7 Application Administration 101

CHAPTER 1

About This Book

This book describes the tools and techniques that system administrators and website managers perform to deploy WebObjects applications. The WebObjects Deployment package allows you to deploy applications developed with the WebObjects Development package, so that they can be accessed through a Web server. You need a WebObjects deployment license to deploy WebObjects applications

The book is intended primarily for system administrators. Application developers can also benefit from the information it provides but it's not required reading for them. This edition of the book is intended for WebObjects 5.0.

Why Read This Book

To deploy WebObjects applications and to administer a deployment, you need to become acquainted with the deployment model of WebObjects. This book shows you how your Web server interacts with the elements of a WebObjects deployment. It also explains what measures you should take to increase your site's performance.

WebObjects Deployment provides tools for most of the tasks you need to accomplish on a regular basis to maintain your site. If you prefer doing things manually, you can use the command line to start individual application instances or the deployment tools themselves.

This book has the following chapters:

- "Introduction to WebObjects Deployment" (page 17) gives you an overview of the deployment approach taken with WebObjects 5. In addition, it lists the ways in which WebObjects Deployment helps you to maintain a secure site.
- "Installing the Software" (page 29) explains which WebObjects Deployment components need to be installed on a computer, taking into account the computer's purpose in your site.
- "HTTP Adaptors" (page 35) describes the function of the HTTP adaptor in your site. It also describes how you customize the adaptors included in WebObjects Deployment if the default configuration does not suit your needs. State discovery is how the HTTP adaptor keeps track of the application instances of your site. The chapter describes the different ways that the adaptor can obtain that information and how you configure the adaptor to use one of those methods.
- "Managing Application Instances" (page 59) introduces you to the deployment tools you use to configure and maintain your site. It also describes the mechanism used in WebObjects to ensure that application instances are always running, helping you maximize your site's up time.
- "Deployment Tasks" (page 67) explains how to perform configuration and maintenance tasks on your site. It also shows how to maintain multiple sites using the same hardware.
- "Application Administration" (page 101) shows you how to monitor and improve your site's performance.
- "Deployment Settings Reference" (page 111) lists the deployment properties you can use to customize your site.
- "Special Deployment Issues" (page 123) lists issues to keep in mind when deploying Java Client applications. It also tells you what to do if you want to deploy WebObjects 4.5.1 applications together with WebObjects 5 applications.

Getting More Information

To get an overview of the WebObjects platform, you should read the book *WebObjects Overview*. You can find general information about WebObjects at http://developer.apple.com/webobjects.

These are other WebObjects documents that provide helpful information:

- *Installation Guide* includes system requirements, compatibility information, and installation instructions. (The *Installation Guide* is included in your WebObjects box.)
- *Post-Installation Instructions* lists tasks that you must perform after installing WebObjects Deployment to confirm that the installation was successful and to tailor it for your specific needs.

What You Should Know

This book assumes you have a background in system administration. You must be familiar with the operation of your platform, especially how to use its command shell editor to issue commands. You must also be acquainted with the operation of your Web server software and TCP/IP networking. Knowledge of WebObjects application development is helpful but not required.

Operating Systems Supported

The following are the supported deployment platforms for WebObjects 5:

- **Mac OS X Server:** Mac OS X Server 10.0.4.
- **Solaris:** Sun Solaris 2.8.
- **Windows 2000:** Microsoft Windows 2000 Pro with J2SE 1.3.1.

WebObjects applications can be deployed on other platforms running J2SE 1.3 or later.

Introduction to WebObjects Deployment

This chapter introduces the essential concepts and tools you use when you deploy WebObjects applications.

The chapter addresses the following topics:

- "The WebObjects Deployment Model" (page 17) introduces you to the WebObjects way of deploying applications. It explains how the users of your applications users send requests to application instances running on your site and how responses (Web pages) are generated and sent back to users.
- "The WebObjects Deployment Environment" (page 21) describes the functions of several elements (both in your platform and in WebObjects Deployment) in a site.
- "Keeping Your Site Secure" (page 26) lists the security-minded features available in WebObjects Deployment.

The WebObjects Deployment Model

A WebObjects deployment has six major parts:

- **Client:** Web browser or Java Client application (A Java Client application).
- **Web server:** application that receives HTTP requests from clients and sends responses back to them.
- **HTTP adaptor:** application that serves as an interface between your Web server and your application instances. The HTTP adaptor routes requests from the Web server to the appropriate instance and sends the responses generated back to the

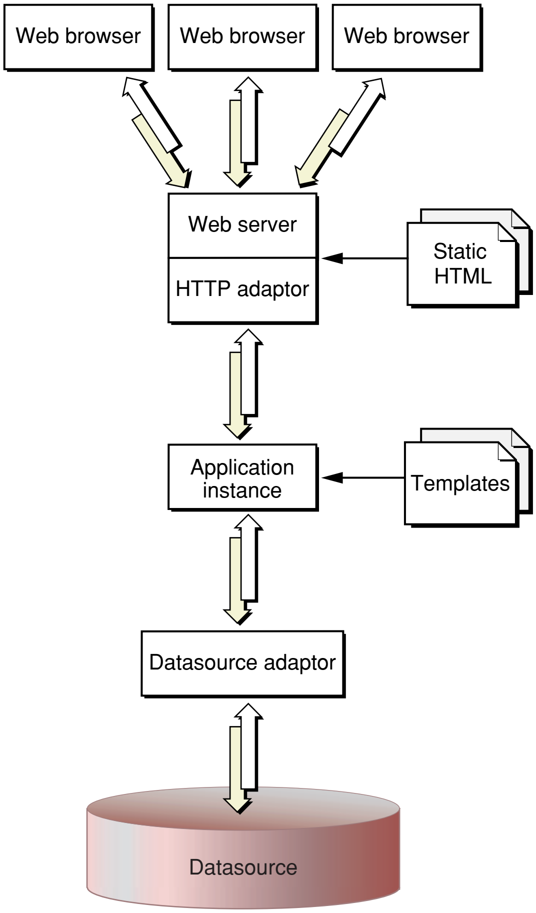
Web server. The adaptor does this while performing **load balancing** to distribute an application's users among its active instances. Load balancing helps to spread the user load of your site evenly across your application hosts.

- **Application instances:** individual processes that receive requests from the HTTP adaptor and send responses back to it. To create a response, an instance can perform calculations, or save or retrieve data from a datasource.
- **Datasource adaptor:** interface between an application instance and your datasource. WebObjects includes a **JDBC** (Java Database Connectivity) adaptor, allowing your applications to connect to any JDBC-compliant database.

  For JDBC connectivity, your database needs a JDBC driver, which you obtain from your database vendor. WebObjects applications can connect to databases that use Type 2 (partly Java) or Type 4 (all Java) JDBC drivers. The JDBC adaptor included with WebObjects Deployment has been certified to work adequately with Type 4 drivers. Type 2 drivers may require special configuration for them to work properly with the adaptor. If your database provides a Type 2 driver, consult with your database vendor to determine how it needs to be configured to work properly with a JDBC adaptor.
- **Datasource:** the mechanism that your applications use to store persistent data. Consult with your datasource vendor to obtain configuration and optimization details.

When an application user sends a request through a Web browser to your Web server, the server forwards the request to the HTTP adaptor. The adaptor then determines which application instance should process the request and forwards the request to it. When the application instance receives the request, it performs all the necessary processing to produce a response (a new Web page). The instance then sends the response page to the adaptor, which forwards it to the Web server. The Web server then forwards the response page to the user's Web browser. This process is illustrated in Figure 2-1.

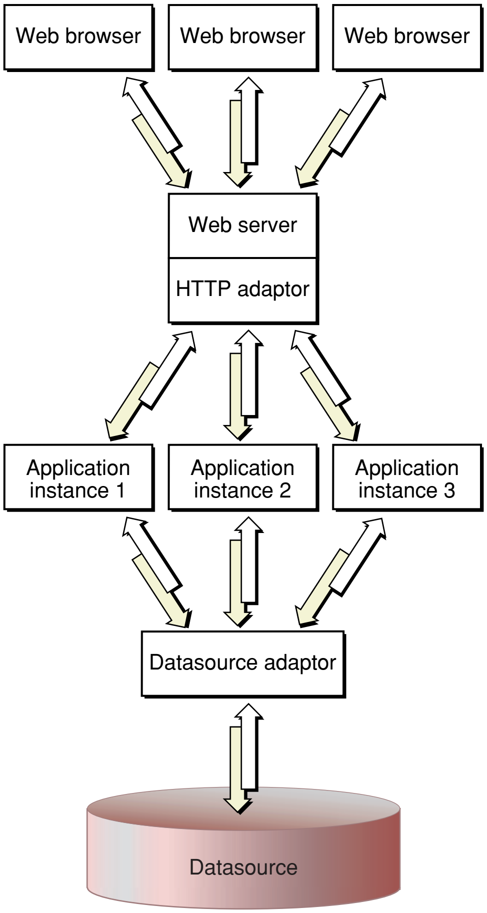
Figure 2-1 WebObjects deployment model



Notice that both the application instance and the Web server contribute to the response page's content. The instance uses templates and logic to generate the HTML code for dynamic pages, while the Web server provides the content of images contained in those pages. The server can also dispense static pages.

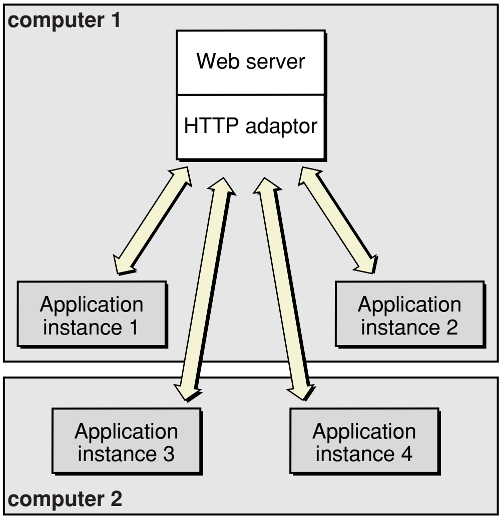
The number of instances of your application necessary to support its users depends on the number of users that connect to your application concurrently. In some cases a single instance is adequate. When one instance is not able to process requests in a timely manner, additional instances can solve the problem. This way, the amount of user-state information that a single instance stores is reduced. In addition, with less state to keep track of, an instance can process requests faster. Figure 2-2 shows a site with one host running multiple instances of an application.

Figure 2-2 WebObjects deployment model—multiple instances of an application



However, adding instances of your application to a host may not be the most effective solution. Eventually a point of diminishing returns will be reached, where adding instances actually decreases your application's performance. In such a case, you should consider adding additional application hosts that run the extra instances required to handle the increased traffic to your site. Figure 2-3 shows how a site with two computers, one acting as a Web server and application host, and the other just as an application host would look.

Figure 2-3 Deployment using two computers



The WebObjects Deployment Environment

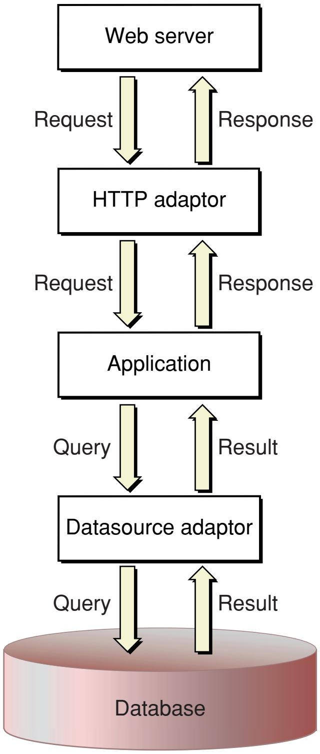
You need to master two important aspects of WebObjects Deployment: the communication paths of client requests and server activity, and the deployment tools you use to configure your site.

Communication Paths

Communication among the elements that make up a deployment occurs in two paths: the data path and the control path.

A client HTTP request takes the data path after it reaches your Web server. Figure 2-4 shows how an HTTP request that your Web server receives is passed to the elements that generate the response.

Figure 2-4 The data path of a WebObjects deployment



Monitor requests take the control path to propagate configuration changes to application hosts and, ultimately, application instances. These include adding application instances, and starting and stopping instances according to a schedule that you define. The HTTP adaptor can obtain site information by polling **wotaskd** processes or by reading the adaptor configuration file. (See "Deployment Tools" (page 23) for information about Monitor and wotaskd.) Figure 2-5 shows the control path.

Figure 2-5 The control path of a WebObjects deployment

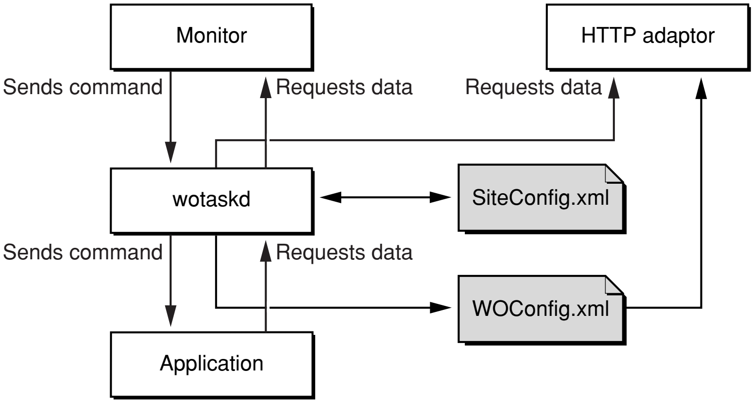
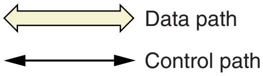


Figure 2-6 shows how the data path and control path are differentiated in the rest of the book.

Figure 2-6 The symbols used to represent the data path and the control path



Deployment Tools

The main tools you use to manage your site are wotaskd and Monitor. Normally, one wotaskd process runs on each application host. If you want to concurrently deploy multiple sites on the same hardware, you can configure a computer to run more than one wotaskd process. This essentially provides you with several independent application hosts per computer.

You manage a group of application hosts using Monitor, a tool that uses your Web browser as its user interface. Monitor lets you set, among other things, instance scheduling and the load-balancing algorithm to be used for each application. Because each Monitor process maintains state information locally, you must run only one instance of Monitor per site. Figure 2-7 shows two application sites on one computer.

Figure 2-7 Two sites deployed on one computer

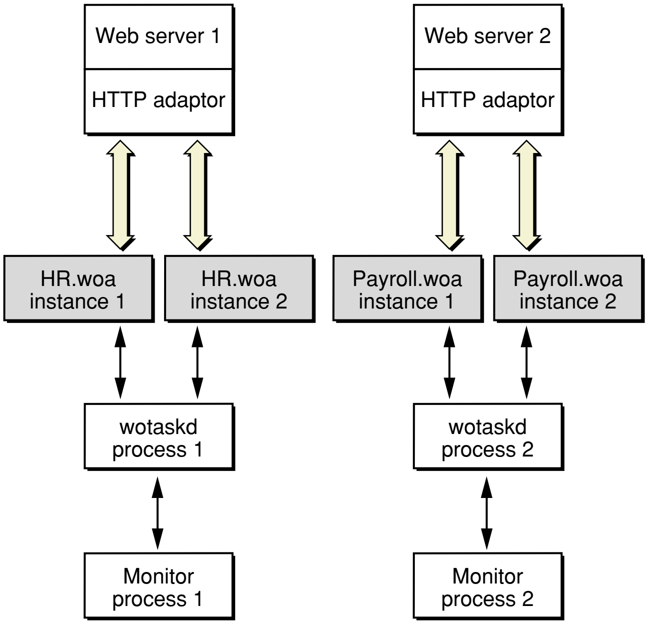
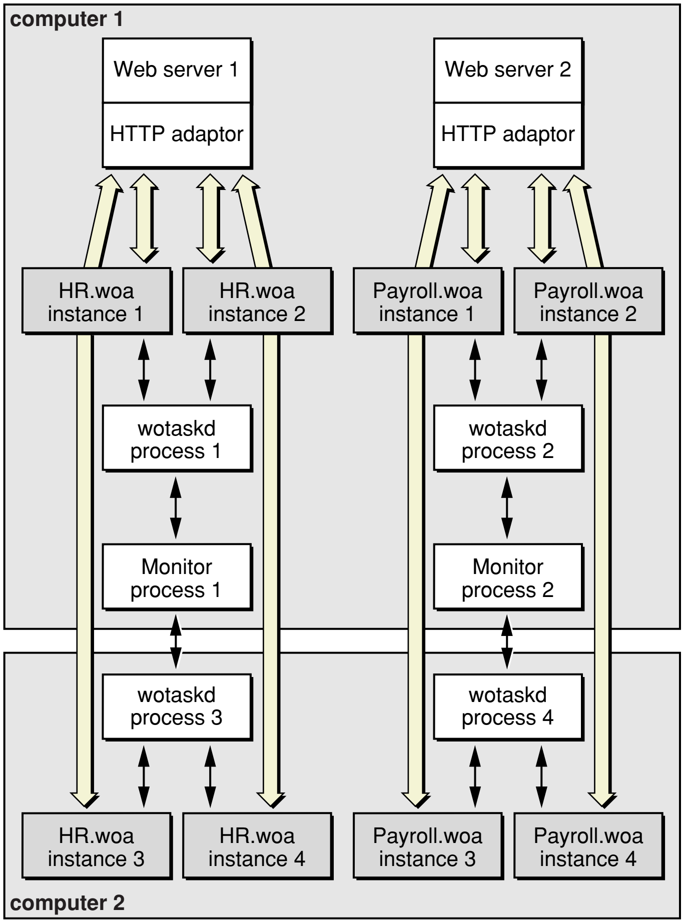


Figure 2-8 shows how you can distribute application instances among two computers.

Figure 2-8 Two sites deployed on two computers



After you configure your site using Monitor, it enforces that configuration by performing tasks such as stopping and restarting application instances according to a schedule you set, and sending email notifications when problems arise. The HTTP adaptor performs load balancing across the instances of each application on your site.

For detailed information on the subjects introduced above, see the following chapters or sections:

- "HTTP Adaptors" (page 35) shows you the different ways in which you can configure the WebObjects adaptor.
- "Deployment Tasks" (page 67) describes how you use Monitor to configure your site.
- "Setting Up Hosts" (page 68) describes how you use Monitor to add application hosts to your site.
- "Configuration Files" (page 59) shows you how the configuration you define in Monitor is distributed among the application hosts of your site.
- "wotaskd Processes" (page 63) explains how wotaskd processes communicate with and manage application instances.
- "Lifebeats" (page 61) explains how application instances communicate with a wotaskd process.
- "Deploying Multiple Sites" (page 98) explains how to configure your platform to deploy multiple sites concurrently.
- "Load Balancing" (page 97) describes how load balancing works and lists the algorithms that the HTTP adaptor can use to implement it.

Keeping Your Site Secure

In a WebObjects deployment, you have several features at your disposal to enhance the security of your site:

- split-installation of applications (application files and Web server resources). By installing application-related files in two locations, you can put sensitive information (such as business logic) into protected locations. Nonsensitive resources (such as image files) can be installed on the Web server's `Document Root` directory. For more information, see "Installing Applications" (page 74).
- restricted access to deployment tools. "Monitor Password" (page 95) explains how you can password-protect access to Monitor and wotaskd through a single page.

- restricted access to development application instances. If your computing environment supports both the development and deployment of applications through the same Web server, access of development instances is restricted by the HTTP adaptor. See “Viewing a Host’s Configuration” (page 72) for details
- restricted access to application instance statistics. Agents external to your organization can use the statistics that your application instances produce to get privileged information. To avoid this, access to the instance statistics page is restricted. See “Setting a Password for the Instance Statistics Page” (page 90) for details.

Installing the Software

This chapter provides an overview of WebObjects Deployment installation issues, including what elements of the software need to be installed on a computer, taking the computer's purpose into account. However, it does not give platform-specific instructions. For detailed installation instructions, see the *Installation Guide* included with your package.

The following topics are addressed:

- "Choosing What to Install" (page 29)
- "Environment Variables in UNIX Platforms" (page 31)
- "Confirming That wotaskd Is Active" (page 32)
- "Default Adaptor Installation" (page 33)
- "Building Adaptors From Source" (page 34)

Choosing What to Install

When you perform a complete installation of WebObjects Deployment on a computer, two types of files are copied to its hard disk: adaptor files and deployment files.

Adaptor Files

These are the files that allow your Web server to communicate with WebObjects application instances. The executable files for the adaptors are placed in the /System/Library/WebObjects/Adaptors directory.

The source files for the adaptors are placed in the /Developer/Examples/WebObjects/Source/Adaptors directory. Also included are HTML files containing instructions for building and installing each adaptor. The code of the adaptors is open source, so you can install them anywhere.

Deployment Files

These files are divided into two groups:

- **Runtime environment.** The runtime environment of WebObjects is implemented in **framework** (.framework) files, whose purpose is similar to dynamically linked library (DLL) files in Windows. Framework files are installed in the /System/Library/Frameworks directory. These frameworks are used by any WebObjects application, including the deployment tools of WebObjects
- **Deployment tools.** WebObjects Deployment includes two deployment tools you use to configure and monitor your site. The files that make up these tools are placed in the /System/Library/WebObjects/JavaApplications directory. Table 3-1 shows the purpose of each tool. For more information on the deployment tools, see “Managing Application Instances” (page 59)

Table 3-1 The WebObjects deployment and administration tools

Filename	Application name	Purpose
JavaMonitor.woa	Monitor	Site configuration and administration
wotaskd.woa	wotaskd	Instance management

Note: Installing WebObjects Deployment on a computer requires a WebObjects Deployment license. Please read the license agreement before installing this package.

Types of WebObjects Deployment Installations

Depending on the purpose of the computer you're installing the software on, there are three types of WebObjects Deployment installations you can perform:

- **Web server only.** On a computer that you want to use as the Web server computer but on which you do not intend to run application instances (including the deployment tools), you need to install only the HTTP adaptor files.
- **Application host only.** On computers that you intend to use only as application hosts, you need to install only the deployment tools. (If you do not plan to run Monitor on that machine, you can delete its files.)
- **Web server and application host.** When one computer can satisfy all your deployment needs or when you want a Web server machine to also run application instances, you need to install the adaptor files and the deployment tools.

Note: On Windows 2000 machines it's not possible to perform tailored WebObjects Deployment installations.

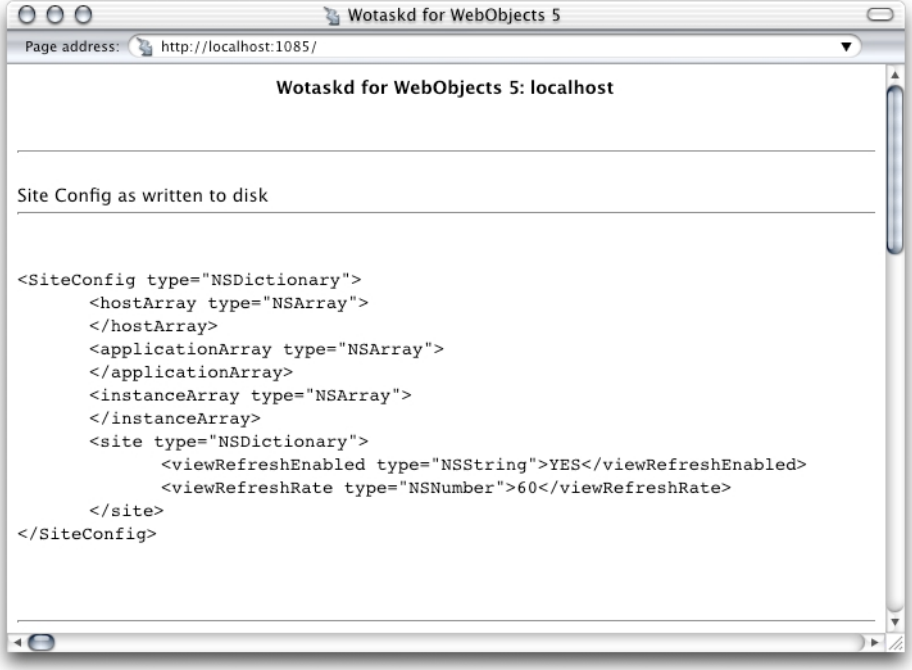
Environment Variables in UNIX Platforms

On Solaris and other UNIX platforms, you must set the value of the `NEXT_ROOT` environment variable. It must point to the directory into which WebObjects Deployment is installed.

Confirming That wotaskd Is Active

After the installation is complete (and if you chose to start **WOServices** in Solaris), a wotaskd process should be running on your computer. (WOServices ensures that wotaskd is running at all times.) To confirm that wotaskd is running, launch a Web browser and enter this address: `http://localhost:1085`. You should see a page like the one in Figure 3-1. For more information, see "Viewing a Host's Configuration" (page 72).

Figure 3-1 The configuration page of a wotaskd process



Wotaskd for WebObjects 5: localhost

Site Config as written to disk

```
<SiteConfig type="NSDictionary">
        <hostArray type="NSArray">
        </hostArray>
        <applicationArray type="NSArray">
        </applicationArray>
        <instanceArray type="NSArray">
        </instanceArray>
        <site type="NSDictionary">
                <viewRefreshEnabled type="NSString">YES</viewRefreshEnabled>
                <viewRefreshRate type="NSNumber">60</viewRefreshRate>
        </site>
</SiteConfig>
```

If a wotaskd process isn't running, these are possible reasons:

- The version of J2SE is not 1.3 or later.

  Upgrade to J2SE 1.3 (version 1.3.1 on Windows 2000).
- Port `1085` is already in use.

  Change the port of the process that's using port `1085`.
- The `NEXT_ROOT` environment variable is not set.

  In platforms other than Mac OS X Server and Windows 2000, make sure that the `NEXT_ROOT` environment variable is set for the user under which you're trying to run wotaskd. This also applies to application instances.

Default Adaptor Installation

Depending on your deployment platform, several HTTP adaptors are installed by default. Table 3-2 lists the adaptors that are installed on each platform.

Table 3-2 The adaptors installed in each platform

	Apache	CGI	ISAPI	NSAPI
Mac OS X Server	x	x		
Solaris	x	x		x
Windows 2000		x	x	x

In Mac OS X Server and Solaris, the Apache adaptor is active by default. Requests in the form `http://.../cgi-bin/WebObjects/` are handled by the Apache adaptor. If you disable the Apache module, then such requests are handled by the CGI adaptor.

To install the ISAPI adaptor, copy the WebObjects.dll file to your Web server's scripts directory. The ISAPI adaptor is loaded into the Web server the first time a request of the form http://.../scripts/WebObjects.dll/ is received. It then remains active until the server is stopped.

Building Adaptors From Source

If you want to use a different HTTP adaptor from the one installed by default on your platform, you can find building and installation instructions for all the adaptors included in WebObjects Deployment in /Developer/Examples/WebObjects/Source/Adaptors/InstallationInstructions.html. To build adaptors from the source code provided, refer to BuildingInstructions.html. in the same directory.

HTTP Adaptors

This chapter provides detailed information about the HTTP adaptors that are included in a WebObjects Deployment installation. The HTTP adaptor is an important piece of an application site. It sits between your Web server and your application instances. It forwards requests from the Web server to the appropriate application instance and responses from the instance back to the Web server. It also performs load balancing between instances of an application.

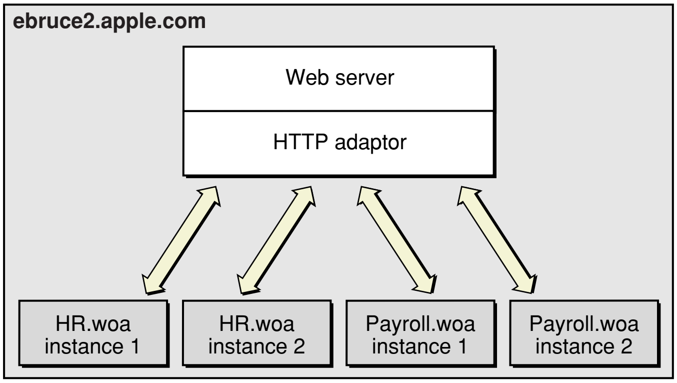
This chapter addresses the following topics:

- "Adaptors, Applications, and Hosts" (page 36) provides a high-level view of the interaction between the HTTP adaptor, application instances, and application hosts.
- "Types of Adaptors" (page 38) explains the differences among the two types of HTTP adaptors that you can use on your site.
- "State Discovery" (page 39) describes how to configure the HTTP adaptor to obtain your site's state dynamically or using a configuration file. It also explains how to change a dynamic configuration into a static one.
- "The WebObjects Adaptor Information Page" (page 51) shows an example of the Web browser page that displays information about an HTTP adaptor.
- "Overriding Default Configuration Settings" (page 53) summarizes all the settings available for HTTP adaptors.

Adaptors, Applications, and Hosts

A WebObjects adaptor forwards requests from a Web server to application instances and returns responses from instances back to the server. You may need to have more than one instance of a given application to support a large number of concurrent users. Figure 4-1 illustrates a simple site, implemented with one computer. It serves two applications, with two instances for each application.

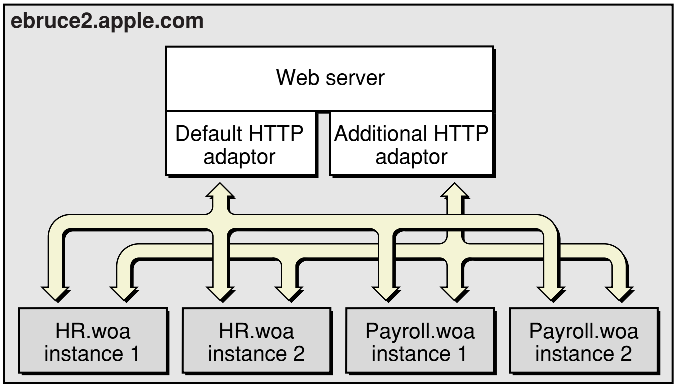
Figure 4-1 Deployment on one computer, using one adaptor



Although the WebObjects installation provides several adaptors, only one is active by default (see "Default Adaptor Installation" (page 33) for details). However, an application can communicate with an adaptor other than the active adaptor.

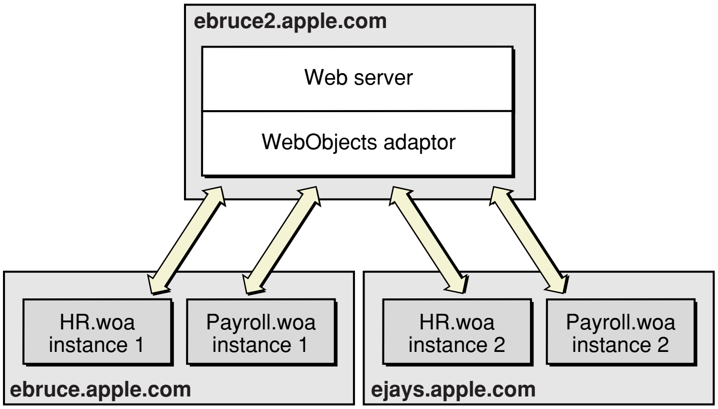
Figure 4-2 depicts an application site running on one machine, using two adaptors.

Figure 4-2 Deployment on one computer using two adaptors



Most sites require multiple computers to ensure that an instance of a particular application is always available. In this kind of deployment, usually one computer runs the Web server and the WebObjects adaptor, while one or more additional machines serve as application hosts. Figure 4-3 illustrates an application site using three computers, one running the Web server and the adaptor, and the other two running application instances.

Figure 4-3 Deployment using three machines using one adaptor



The HTTP adaptor needs to periodically determine your site's state—which application instances are running. There are two ways in which the adaptor can obtain this information:

- **Dynamically:** The adaptor determines your site's state by asking each application host for its state. The adaptor can use a multicast request to find out which hosts are available or you can define a host list for it. Using this method, you avoid having to configure new hosts as you add them to your site.
- **Statically:** An adaptor configuration file contains host and application information about your site; it includes information about every application instance you want to run. After the adaptor reads the file, it has all the information it needs to communicate with the application instances you want to run. Using this kind of configuration avoids multicast requests and host polling. However, when you add new hosts, you'll have to update the configuration file. For more information on the adaptor configuration file, see "Using a Configuration File" (page 44).

You configure your site using Monitor, a Web-browser–based tool.

Types of Adaptors

There are two general types of HTTP adaptors, **CGI adaptors** and **API-based adaptors**. CGI adaptors are portable across many platforms. API-based adaptors are generally more efficient than CGI adaptors.

CGI Adaptors

WebObjects Deployment includes a CGI adaptor, which is an executable file named `WebObjects`; in Windows 2000, it's named `WebObjects.exe`. The CGI adaptor resides in the Web server's `cgi-bin` or `scripts` directory. This adaptor works with any Web server that conforms to the CGI standard.

The major drawback of CGI adaptors is their performance. When the Web server receives a request from a browser, it creates a new process for the adaptor. When the adaptor is done processing the request, the process is terminated.

The CGI adaptor is installed by default on all platforms, but it may not be the active one on your platform. See "Default Adaptor Installation" (page 33) for more information.

API-Based Adaptors

API-based adaptors are based on APIs specific to a particular Web server. They allow CGI-like tasks to run as part of the main server process, avoiding the creation and termination of a process for each request. Table 4-1 lists the API-based adaptors included with WebObjects and the platforms on which they are supported.

Table 4-1 API-based adaptors and supported platforms

Adaptor	API	Supported platforms
Apache	Apache's module API	Mac OS X Server Solaris
ISAPI	Microsoft's Internet Information Server API	Windows 2000
NSAPI	Netscape Server 3.5 API	Solaris Windows 2000

State Discovery

Your site's state is represented by

- a list of application hosts
- a list of running application instances on each host

The HTTP adaptor captures your site's state at regular intervals, which you set when you configure the adaptor. You define the method that the adaptor uses to gather state information by configuring the adaptor itself. For details, see "Overriding Default Configuration Settings" (page 53).

The adaptor can obtain the state of your site using one of three methods:

- **Multicast request.** The adaptor sends a multicast request to find out what application hosts are available. After the host list is compiled, the adaptor polls each host to get its list of running application instances
- **Host list.** This method requires that you configure the host list in the adaptor itself. As with the first method, the adaptor polls the hosts on the list for their lists of running application instances.
- **Configuration file.** The adaptor obtains the site's configuration by reading an **XML** (Extensible Markup Language) formatted document.

The method that requires the least administration on your part is the multicast request. If an application host goes down, the adaptor automatically removes the application instances running on it from its list of active instances. When the host is brought back up, the adaptor adds the instances back to its list. You should use this method if your site has many application hosts. See "Using a Multicast Request" (page 41) for more information.

The second method, defining a host list for your adaptor, eliminates the multicast request. Use this method if you do not want the adaptor to send regular multicast requests out on your network or if you seldom add or remove application hosts from your site. This is the method that is active by default. However, the host list contains only one host, `localhost`. For details, see "Using a Defined Host List" (page 44).

In the third method, using a configuration file, the HTTP adaptor obtains your site's configuration by reading a file. This file can be static or it can be dynamically updated as you configure your site with Monitor. For details, see "Using a Configuration File" (page 44).

You can write the adaptor configuration file in one of two ways:

- **Manually.** The information in the configuration file is stored in a structured document using the XML format. For details, see "The HTTP Adaptor Configuration File" (page 44).
- **Using Monitor and wotaskd.** After configuring your site to your liking using Monitor, you can have a file created for you or you can copy and paste the information. See "Creating the HTTP Adaptor Configuration File" (page 47) for more information.

Using a Multicast Request

When you configure an adaptor to obtain your site's state using a multicast discovery request, the adaptor obtains the list of active application hosts by broadcasting a message to which each computer configured as a WebObjects application host responds. After the adaptor compiles the list of available hosts, it polls each one to obtain its state (the list of running application instances).

There are drawbacks to using the multicast method:

- It increases network traffic. By default, the HTTP adaptor send a multicast request every 100 seconds
- A host may become unavailable between discovery requests if the multicast request or a wotaskd process's response is lost (multicast is an inherently unreliable protocol).
- Normally, multicast broadcasts are limited to a subnet. However, you can configure your routers to pass on the multicast request to other subnets if you wish.

By default, wotaskd does not respond to multicast requests. To be able to use the multicast request method, you must configure wotaskd processes in your application hosts to respond to multicast requests.

Multicast Request

To discover available hosts, the adaptor sends a host discovery request on the multicast channel (a nonrouting IP address and a port number), which is set to IP address `239.128.14.2` and port `1085` by default. The frequency of each multicast request is ten times as long as the adaptor's configuration refresh interval. For details on how to change the multicast channel, see "Setting the Multicast Address and Port" (page 53), "WOPort" (page 120), "WOMulticastAddress" (page 122), and "WORespondsToMulticastQuery" (page 122). When a wotaskd process starts, it creates a UDP (User Datagram Protocol) socket that listens to the multicast channel through which it receives multicast requests.

Note: If you change the address and port that adaptors use to send multicast requests, you must also change the address and port that wotaskd processes use to receive multicast requests.

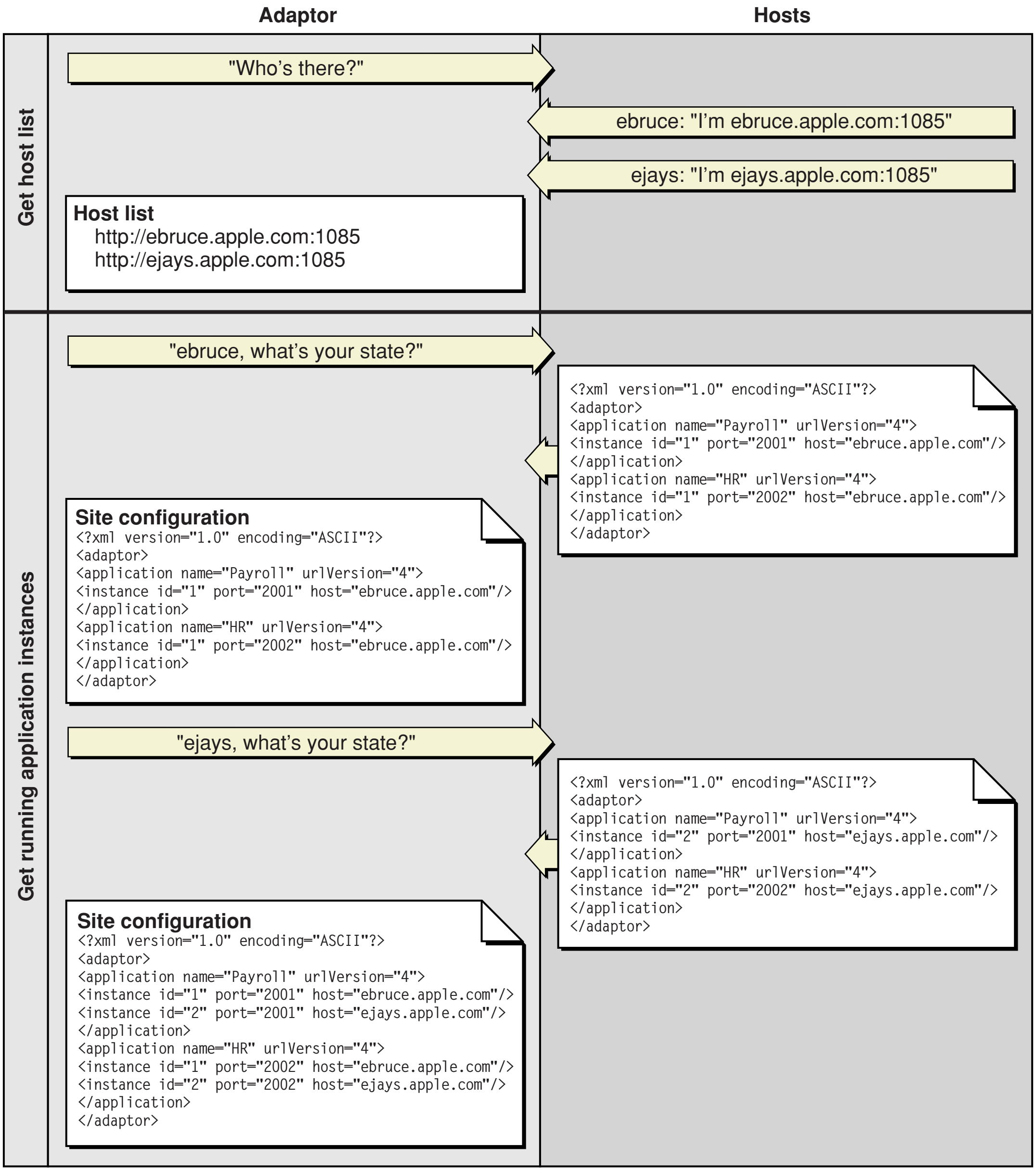
When each wotaskd process receives the multicast request, it replies with its URL, such as `http://host1.site.com:1085`. The adaptor in turn compiles a list of these URLs.

Sending a multicast request on an entire subnet is an expensive procedure. If your available hosts never change, consider using a defined host list instead.

Host Polling

After the HTTP adaptor constructs the host list, it polls each application host on the list for information on the active application instances running on it. Each wotaskd process, in turn, sends its state information using the format in Listing 4-2 (page 45). Host polling to obtain information on active instances occurs at the interval indicated in the configuration refresh interval setting for the HTTP adaptor. Figure 4-4 illustrates the process used to determine the configuration of the site in Figure 4-3 (page 37).

Figure 4-4 Dynamic site configuration using multicast request and polling



Adaptor
Hosts
Get host list
"Who's there?"
ebruce: "I'm ebruce.apple.com:1085"
ejays: "I'm ejays.apple.com:1085"
Host list
http://ebruce.apple.com:1085
http://ejays.apple.com:1085
Get running application instances
"ebruce, what's your state?"
Site configuration
"ejays, what's your state?"
Site configuration

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
<application name="Payroll" urlVersion="4">
<instance id="1" port="2001" host="ebruce.apple.com"/>
</application>
<application name="HR" urlVersion="4">
<instance id="1" port="2002" host="ebruce.apple.com"/>
</application>
</adaptor>
```

Site configuration

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
<application name="Payroll" urlVersion="4">
<instance id="1" port="2001" host="ebruce.apple.com"/>
</application>
<application name="HR" urlVersion="4">
<instance id="1" port="2002" host="ebruce.apple.com"/>
</application>
</adaptor>
```

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
<application name="Payroll" urlVersion="4">
<instance id="2" port="2001" host="ejays.apple.com"/>
</application>
<application name="HR" urlVersion="4">
<instance id="2" port="2002" host="ejays.apple.com"/>
</application>
</adaptor>
```

Site configuration

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
<application name="Payroll" urlVersion="4">
<instance id="1" port="2001" host="ebruce.apple.com"/>
<instance id="2" port="2001" host="ejays.apple.com"/>
</application>
<application name="HR" urlVersion="4">
<instance id="1" port="2002" host="ebruce.apple.com"/>
<instance id="2" port="2002" host="ejays.apple.com"/>
</application>
</adaptor>
```

Using a Defined Host List

This method is similar to the one described in "Using a Multicast Request" (page 41). The only difference is that the HTTP adaptor skips the first part, the multicast request. The host polling process occurs at the interval set in the adaptor's configuration refresh interval setting.

You must explicitly define a host list for each adaptor. See "Setting the Host List" (page 54) for details on defining the host list for each of the adaptors provided.

Using a Configuration File

Using an HTTP adaptor configuration file is useful when you want to have a static site configuration (one in which application instances are not stopped after they are started) or if you want to use Monitor to configure your site and have the adaptor read your configuration changes immediately. (The adaptor reads the configuration file every 10 seconds to determine which application instances are active.)

This method also provides a way of having more than one configuration of your site available. You can switch among different configurations by placing the appropriate configuration file in the configuration directory.

"The HTTP Adaptor Configuration File" (page 44) explains how the file is structured and lists the properties that it defines. For instructions on creating the configuration file and configuring the HTTP adaptor to use it, see "Creating the HTTP Adaptor Configuration File" (page 47).

The HTTP Adaptor Configuration File

You can set up the HTTP adaptor to get your site's configuration by reading an HTTP adaptor configuration file (called `WOConfig.xml` by default) in the configuration directory (`/Library/WebObjects/Configuration` by default). You should have only one adaptor configuration file per Web server so that it can perform load balancing effectively. (See "Load Balancing" (page 97) for details.) In addition, in a site with multiple Web servers, if two servers share the configuration file, instead of deploying two sites you would be deploying the same site twice. Listing 4-1 shows a configuration file that defines a site with two application hosts (`ebruce.apple.com` and `ejays.apple.com`), each running two application instances, one of the Payroll application and the other of the HR application.

Listing 4-1 A WebObjects adaptor configuration file

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
  <application name="Payroll" urlVersion="4">
    <instance id="1" port="2002" host="ebruce.apple.com"/>
    <instance id="2" port="2001" host="ejays.apple.com"/>
  </application>
  <application name="HR" urlVersion="4">
    <instance id="1" port="2001" host="ebruce.apple.com"/>
    <instance id="2" port="2002" host="ejays.apple.com"/>
  </application>
</adaptor>
```

The HTTP adaptor configuration file provides the HTTP adaptor with information about your site's registered application instances. The format of the configuration file is provided in Listing 4-2 (you can also view it by opening the `woadaptor.dtd` file, located in the `/Developer/Examples/WebObjects/Source/Adaptors` directory). For information on the properties defined in the configuration file, consult Table 4-2 (page 46).

Listing 4-2 Format of the HTTP adaptor configuration file

```
<?xml version="1.0" encoding="ASCII"?>

<!DOCTYPE WebObjectsAdaptorConfiguration SYSTEM "woadaptor.dtd">

<adaptor>
    <application name=STRING
            retries=NUMBER
            scheduler=["RANDOM"|"ROUNDROBIN"|"LOADAVERAGE"]
            dormant=NUMBER
            protocol="http"
            redir=URL
            poolSize=NUMBER
            urlVersion=["3"|"4"]
            additionalArgs="unspecified"
```

```
        >
        <instance    id=NUMBER  port=NUMBER  host=STRING
            sendTimeout=NUMBER
            recvTimeout=NUMBER
            cnctTimeout=NUMBER
            sendBufSize=NUMBER
            recvBufSize=NUMBER
            additionalArgs="unspecified"
        >
        </instance>
    </application>
</adaptor>
```

Table 4-2 The properties of the HTTP adaptor configuration file

Attribute	
name	This property is used by the adaptor to implement load balancing. The adaptor can load-balance only between instances with the same application name. The property can be used to create groups of instances, even when the instances share the same executable file. This argument is set automatically for instances started by wotaskd.
retries	The number of times a request is retried (trying several instances) if a communications failure occurs before an error page is returned to the Web server.
scheduler	The load-balancing scheme used by the adaptor for instances of the application. The options provided by WebObjects are Round-Robin, Random, and Load Average. You can also use a custom load balancer by choosing the Custom option and entering the load balancer’s name.
dormant	The number of times the adaptor skips an instance of the application before trying again.
redir	The URL that the user is redirected to when an instance fails to respond to a direct request.
poolSize	The maximum number of simultaneous connections the adaptor should keep open for each configured instance.

Table 4-2 The properties of the HTTP adaptor configuration file

Attribute	
`urlVersion`	The WebObjects version to use for URL parsing and formatting. All WebObjects 4, 4.5, and 5 applications use version 4 URLs by default.
`additionalArgs`	Additional information to send to the instance when it's started.
`id`	The instance's identification number. Must be unique for the load-balancing process to operate correctly.
`port`	The port on which the instance runs.
`host`	Specifies the network interface that an instance binds to. This argument should only be used on hosts with multiple network interfaces (IP addresses).
`sendTimeout`	The length of time, in seconds, that the adaptor attempts to send data to an instance of the application before giving up.
`recvTimeout`	The length of time, in seconds, that the adaptor waits for a response from an instance of the application before giving up.
`cnctTimeout`	The length of time, in seconds, before the adaptor gives up connecting to an instance.
`sendBufSize`	The size, in bytes, of the TCP/IP socket send buffer that's used for adaptor-to-instance communication.
`recvBufSize`	The size, in bytes, of the TCP/IP socket receive buffer that's used for adaptor-to-instance communication.

Creating the HTTP Adaptor Configuration File

You can define your site's configuration by writing the HTTP adaptor configuration file by hand. However, Monitor provides you with an easy-to-use interface that facilitates that task.

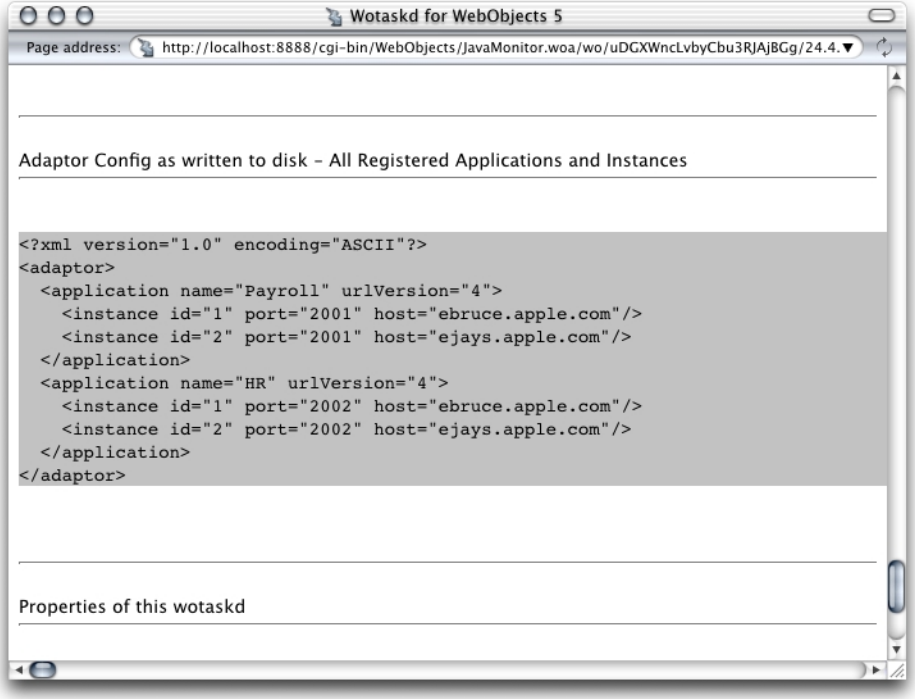
"Deployment Tasks" (page 67) shows you how to configure your site using Monitor. When you are satisfied with your configuration, you can save your settings into a configuration file by copying and pasting or by telling wotaskd to write the file.

To use the copy-and-paste method, follow these steps:

1. In Monitor, display the Hosts page.
2. Click YES for any host.

   The host configuration page is displayed in a new Web browser window.
3. Copy the contents of the section "Adaptor Config as written to disk—All Registered Applications and Instances," as shown in Figure 4-5.

Figure 4-5 Copying the information that makes up the HTTP adaptor configuration file



Wotaskd for WebObjects 5
Page address: http://localhost:8888/cgi-bin/WebObjects/JavaMonitor.woa/wo/uDGXWncLvbyCbu3RJAjBGg/24.4.

Adaptor Config as written to disk – All Registered Applications and Instances

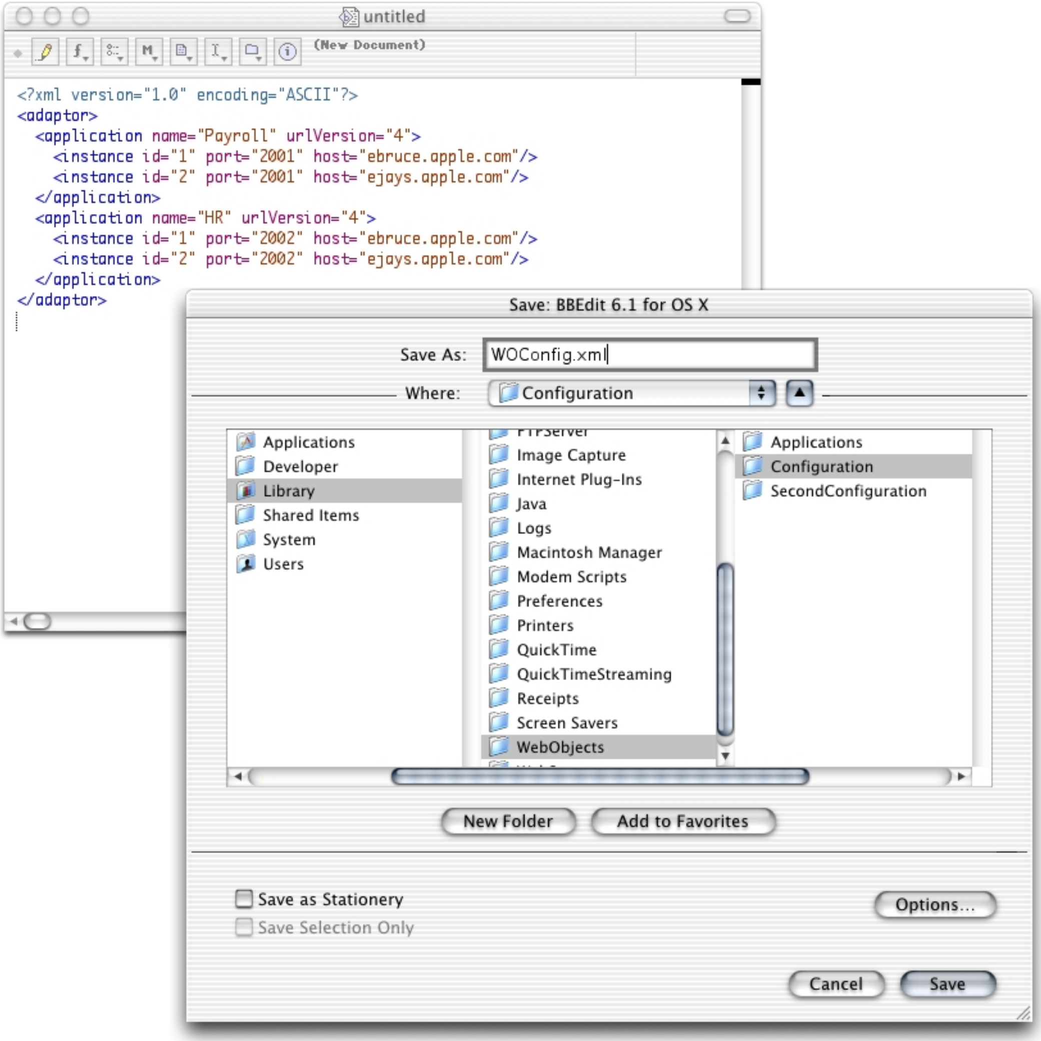
```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
  <application name="Payroll" urlVersion="4">
    <instance id="1" port="2001" host="ebruce.apple.com"/>
    <instance id="2" port="2001" host="ejays.apple.com"/>
  </application>
  <application name="HR" urlVersion="4">
    <instance id="1" port="2002" host="ebruce.apple.com"/>
    <instance id="2" port="2002" host="ejays.apple.com"/>
  </application>
</adaptor>
```

Properties of this wotaskd

4. Using a text editor, create a new file and paste the contents of the clipboard into it.

5. Save the file as `WOConfig.xml` (or any other name you choose) in the configuration directory.

Figure 4-6 Creating and saving the HTTP adaptor configuration file



untitled

(New Document)

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
  <application name="Payroll" urlVersion="4">
    <instance id="1" port="2001" host="ebruce.apple.com"/>
    <instance id="2" port="2001" host="ejays.apple.com"/>
  </application>
  <application name="HR" urlVersion="4">
    <instance id="1" port="2002" host="ebruce.apple.com"/>
    <instance id="2" port="2002" host="ejays.apple.com"/>
  </application>
</adaptor>
```

Save: BBEdit 6.1 for OS X

Save As: WOConfig.xml

Where: Configuration

If instead of copying and pasting you want wotaskd to create the configuration file for you, you can start a wotaskd process specifically to create the file or you can tell wotaskd to continually maintain the configuration file.

To start a wotaskd process specifically to create the file, you must first stop the process that corresponds to the site you configured if it's already running on the Web server computer.

To start a wotaskd process that writes the configuration file to the default location, execute the following two commands using your command shell editor:

```
cd /System/Library/WebObjects/JavaApplications/wotaskd.woa
./wotaskd -WOPort <port> -WOSavesAdaptorConfiguration true
```

To specify a different location for the HTTP adaptor configuration file, follow the instructions in "WODeploymentConfigurationDirectory" (page 121). If you want to give the file a different name, "Setting the HTTP Adaptor Configuration File" (page 55) shows you how.

To tell wotaskd to maintain the configuration file on a permanent basis, add the following to the WOServices script line that starts the wotaskd process:

```
-WOSavesAdaptorConfiguration true
```

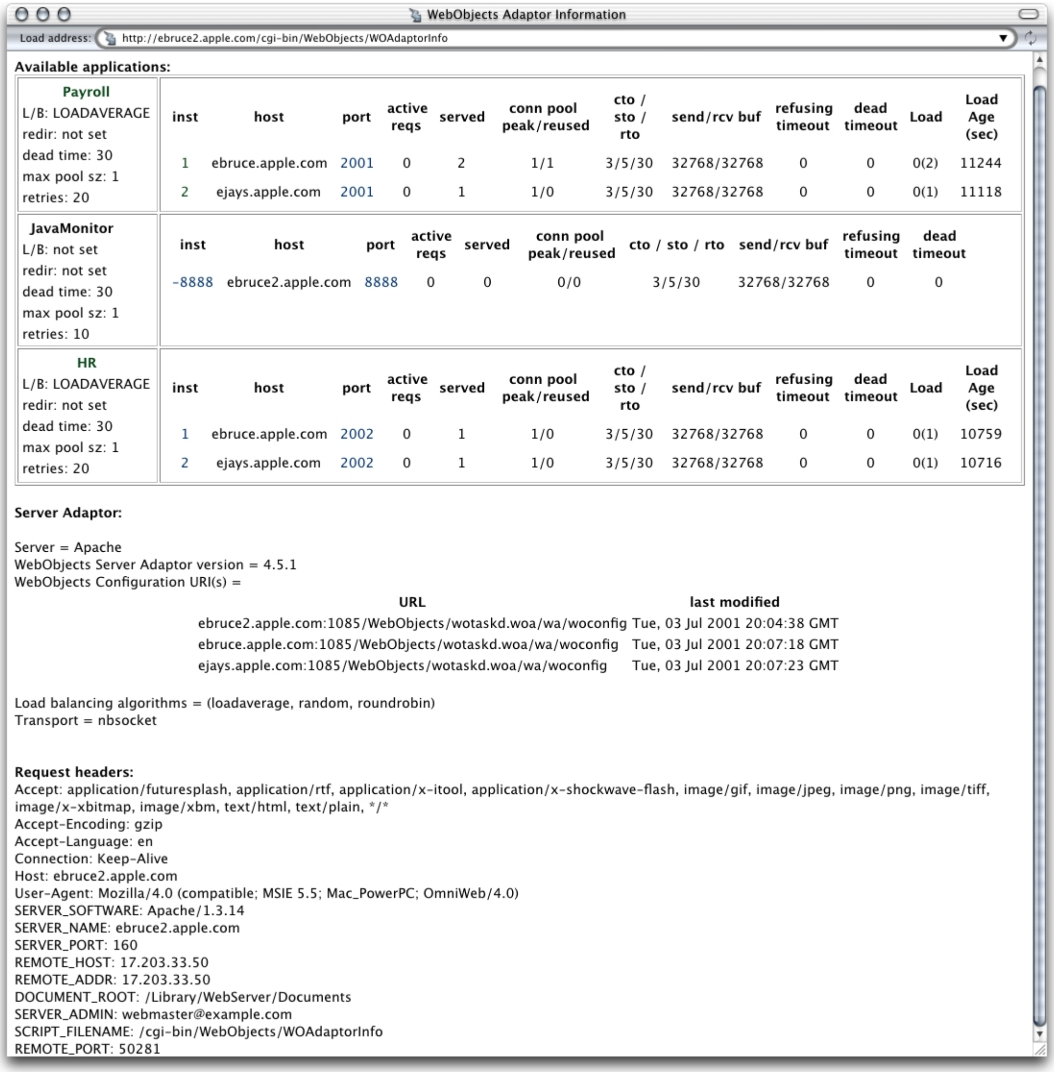
When you restart your Web server, the HTTP adaptor configuration file is updated every time you make a change to your site's configuration through Monitor. The changes are picked up by the HTTP adaptor the next time it reads the configuration file.

To configure the HTTP adaptor to read the configuration file instead of using a multicast request or a host list, follow the instructions in "Setting the HTTP Adaptor Configuration File" (page 55).

The WebObjects Adaptor Information Page

The WebObjects Adaptor Information page displays information about an HTTP adaptor. Access to this page is disabled by default so you must modify the adaptor configuration file to allow access. See "Setting Access to the WebObjects Adaptor Information Page" (page 55) for details. Figure 4-7 shows an example of a WebObjects Adaptor Information page.

Figure 4-7 The WebObjects Adaptor Information page


WebObjects Adaptor Information
Load address: http://ebruce2.apple.com/cgi-bin/WebObjects/WOAdaptorInfo

Available applications:

Payroll
L/B: LOADAVERAGE
redir: not set
dead time: 30
max pool sz: 1
retries: 20

inst	host	port	active reqs	served	conn pool peak/reused	cto / sto / rto	send/rcv buf	refusing timeout	dead timeout	Load	Load Age (sec)
1	ebruce.apple.com	2001	0	2	1/1	3/5/30	32768/32768	0	0	0(2)	11244
2	ejays.apple.com	2001	0	1	1/0	3/5/30	32768/32768	0	0	0(1)	11118

JavaMonitor
L/B: not set
redir: not set
dead time: 30
max pool sz: 1
retries: 10

inst	host	port	active reqs	served	conn pool peak/reused	cto / sto / rto	send/rcv buf	refusing timeout	dead timeout
-8888	ebruce2.apple.com	8888	0	0	0/0	3/5/30	32768/32768	0	0

HR
L/B: LOADAVERAGE
redir: not set
dead time: 30
max pool sz: 1
retries: 20

inst	host	port	active reqs	served	conn pool peak/reused	cto / sto / rto	send/rcv buf	refusing timeout	dead timeout	Load	Load Age (sec)
1	ebruce.apple.com	2002	0	1	1/0	3/5/30	32768/32768	0	0	0(1)	10759
2	ejays.apple.com	2002	0	1	1/0	3/5/30	32768/32768	0	0	0(1)	10716

Server Adaptor:

Server = Apache
WebObjects Server Adaptor version = 4.5.1
WebObjects Configuration URI(s) =

URL	last modified
ebruce2.apple.com:1085/WebObjects/wotaskd.woa/wa/woconfig	Tue, 03 Jul 2001 20:04:38 GMT
ebruce.apple.com:1085/WebObjects/wotaskd.woa/wa/woconfig	Tue, 03 Jul 2001 20:07:18 GMT
ejays.apple.com:1085/WebObjects/wotaskd.woa/wa/woconfig	Tue, 03 Jul 2001 20:07:23 GMT

Load balancing algorithms = (loadaverage, random, roundrobin)
Transport = nbsocket

Request headers:
Accept: application/futuresplash, application/rtf, application/x-itool, application/x-shockwave-flash, image/gif, image/jpeg, image/png, image/tiff, image/x-xbitmap, image/xbm, text/html, text/plain, */*
Accept-Encoding: gzip
Accept-Language: en
Connection: Keep-Alive
Host: ebruce2.apple.com
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Mac_PowerPC; OmniWeb/4.0)
SERVER_SOFTWARE: Apache/1.3.14
SERVER_NAME: ebruce2.apple.com
SERVER_PORT: 160
REMOTE_HOST: 17.203.33.50
REMOTE_ADDR: 17.203.33.50
DOCUMENT_ROOT: /Library/WebServer/Documents
SERVER_ADMIN: webmaster@example.com
SCRIPT_FILENAME: /cgi-bin/WebObjects/WOAdaptorInfo
REMOTE_PORT: 50281

Overriding Default Configuration Settings

For the most part, you shouldn't need to modify the default values of settings in the configuration file. However, if you want to change the way the HTTP adaptor obtains your site's state information, for example, you'll need to perform some of the procedures explained here.

These are the tasks explained in this section:

- "Setting the Multicast Address and Port" (page 53)
- "Setting the Host List" (page 54)
- "Setting the HTTP Adaptor Configuration File" (page 55)
- "Setting Access to the WebObjects Adaptor Information Page" (page 55)
- "Setting an Alias for cgi-bin in the WebObjects URL" (page 56)
- "Setting the Document Root Path of the Web Server" (page 57)

Setting the Multicast Address and Port

The following list explains how to set the multicast address, port, and configuration refresh interval (in seconds) in the supported adaptors. The default values for each of these properties are `239.128.14.2`, `1085`, and `10` respectively. The adaptor uses the configuration interval to determine the amount of time that passes between state discoveries on your site. The host discovery process occurs 10 times less frequently than the time indicated by the configuration refresh interval. With the configuration refresh interval set to `10`, the discovery process occurs every 100 seconds.

- **Apache:** Set the value of the `WebObjectsConfig` variable in the `apache.conf` file to the desired values, using the format shown below:

  ```
  WebObjectsConfig webobjects://<address>:<port> <configuration_interval>
  ```

- **ISAPI:** Add two keys to the Registry, `CONF_URL` and `CONF_INTERVAL`, choose `REG_SZ` as their data type, and set their values as follows:

  ```
  \\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\CONF_U
  RL webobjects://<address>:<port>
  ```

```
\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\CONF_I
NTERVAL <configuration_interval>
```

- **NSAPI:** Add the following line to the `obj.conf` file:

```
Init fn="WebObjects_init" root="/opt/ns-home/docs" config="webobjects://
<address>:<port> confinterval="<configuration_interval>"
```

- **CGI:** Set the `WO_CONFIG_URL` environment variable to `webobjects://<address>:<port>`. Make sure your Web server is configured to pass the variable to the adaptor (consult your Web server's documentation for instructions).

Setting the Host List

The following list explains how to set a host list for a site with two hosts, host1 and host2, in the supported adaptors with a configuration interval of `10` (the configuration interval cannot be set in the CGI adaptor).

- **Apache:** Set the `WebObjectsConfig` variable in the `apache.conf` file to the desired list of hosts. By default it's set to `http://localhost:1085 10` (the 10 is the configuration refresh interval). Separate each host with a comma, as shown in the following example

```
WebObjectsConfig http://host1:1085,http://host2:1085 10
```

- **ISAPI:** Add two keys to the Registry, `CONF_URL` and `CONF_INTERVAL`, choose `REG_SZ` as their data type, and set their values as follows

```
\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\CONF_U
RL http://host1:1085,http://host2:1085

\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\CONF_I
NTERVAL 10
```

- **NSAPI:** Set the `WebObjects_init` function's arguments in the `obj.conf` file as follows

```
Init fn="WebObjects_init" root="/opt/ns-home/docs" config="http://
host1:1085,http://host2:1085" confinterval="10"
```

- **CGI:** Set the `WO_CONFIG_URL` environment variable to `http://host1:1085,http://host:1085`. Make sure the Web server is configured to pass the variable to the adaptor (consult your Web server's documentation for instructions)

Setting the HTTP Adaptor Configuration File

- **Apache:** Set the value of `WebObjectsConfig` variable in the `apache.conf` file to the path of the adaptor configuration file.

  ```
  WebObjectsConfig file://<path-to-an-xml-config-file> 10
  ```

- **ISAPI:** Add the `CONF_URL` key to the Registry, choose `REG_SZ` as its data type, and set the adaptor configuration file path as its value, as the following example shows:

  ```
  \\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\CONF_U
  RL file://<path-to-an-xml-config-file>
  ```

- **NSAPI:** Set the arguments of the `WebObjects_init` function in the `obj.conf` file as follows:

  ```
  Init fn="WebObjects_init" root="/opt/ns-home/docs" config="file://
  <path-to-an-xml-config-file>"
  ```

- **CGI:** Set the `WO_CONFIG_URL` environment variable to `file://<path-to-an-xml-config-file>`. Make sure the Web server is configured to pass the variable to the adaptor (consult your Web server's documentation for instructions).

Setting Access to the WebObjects Adaptor Information Page

You can provide access to the WebObjects adaptor information page to a specific user or to everyone. To provide access to a single user, you set the values of the `username` and `password` attributes. To provide public access, set the `username` attribute to `public`. The following list explains how to provide access to the information page to a user named joe in the supported adaptors. For the changes to take effect, you need to restart the Web server.

- **Apache:** Add the following lines to the `apache.conf` file, located in the `/System/Library/WebObjects/Adaptor/Apache` directory:

  ```
  WebObjectsAdminUsername joe
  WebObjectsAdminPassword secret
  ```

- **ISAPI:** Add two keys to the Registry, `WOUSERNAME` and `WOPASSWORD`, choose `REG_SZ` as their data type, and set their values as follows:

```
\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\WOUSER
NAME joe

\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\WOPASS
WORD secret
```

- **NSAPI:** Add the following line to `obj.conf`:

  ```
  Init fn="WebObjects_init" root="/opt/ns-home/docs" config="webobjects://
  <address>:<port>" username="joe" password="secret"
  ```

- **CGI:** Set the `WO_ADAPTOR_INFO_USERNAME` and `WO_ADAPTOR_INFO_PASSWORD` environment variables to the appropriate values. Make sure the Web server is configured to pass the variables to the adaptor (consult your Web server's documentation for instructions).

Setting an Alias for cgi-bin in the WebObjects URL

The following list explains how to change the `cgi-bin` part of the URL used to connect to an application instance to `Store` in the Apache, NSAPI, and CGI adaptors:

- **Apache:** In the `apache.conf` file, change the line

  ```
  WebObjectsAlias /cgi-bin/WebObjects
  ```

  to

  ```
  WebObjectsAlias /Store/WebObjects
  ```

- **NSAPI:** In the `obj.conf` file, change the lines

  ```
  NameTrans from="/cgi-bin/WebObjects" fn="WebObjectsNameTrans"
  name="webobjects"
  NameTrans from="/cgi-bin" fn="pfx2dir" dir="/opt/ns-home/cgi-bin"
  name="cgi"
  ```

  to

  ```
  NameTrans from="/Store/WebObjects" fn="WebObjectsNameTrans"
  name="webobjects"
  NameTrans from="/Store" fn="pfx2dir" dir="/opt/ns-home/cgi-bin"
  name="cgi"
  ```

Setting the Document Root Path of the Web Server

- **Apache:** In the `apache.conf` file, change the line

  ```
  WebObjectsDocumentRoot /Library/WebServer/Documents
  ```

  to

  ```
  WebObjectsDocumentRoot <document-root-path>
  ```

- **ISAPI:** Add the following Registry entry:

  ```
  \\HKEY_LOCAL_MACHINE\\SOFTWARE\\Apple\\WebObjects\\Configuration\\DOCUME
  NT_ROOT <document-root-path>
  ```

- **NSAPI:** In the `obj.conf` file, change the value of the `root` variable in every line that defines it to the desired path. For example, the line

  ```
  Init fn="WebObjects_init" root="/opt/ns-home/docs" config="http://
  localhost:1085"
  ```

  needs to be changed to

  ```
  Init fn="WebObjects_init" root="<document-root-path>" config="http://
  localhost:1085"
  ```

- **CGI:** Set the value of the `CGI_DOCUMENT_ROOT` environment variable to the desired path. Make sure that your Web server is configured to pass the variable to the adaptor (consult your Web server's documentation for instructions).

CHAPTER 5

Managing Application Instances

This chapter provides a detailed description of Monitor and wotaskd, two tools you use to manage the application instances running on your site, and it explains how instances communicate with wotaskd:

- "Configuration Files" (page 59) provides an overview of the configuration files that can be used in your site.
- "Lifebeats" (page 61) introduces lifebeats, the mechanism used by wotaskd processes to determine the status of the application instances that they manage.
- "wotaskd Processes" (page 63) explains how wotaskd processes ensure that the application instances they manage are always running. It also describes how to restrict access to wotaskd using Monitor.
- "Starting WebObjects Services" (page 63) shows you how to configure a computer to start wotaskd and Monitor during system startup.

Configuration Files

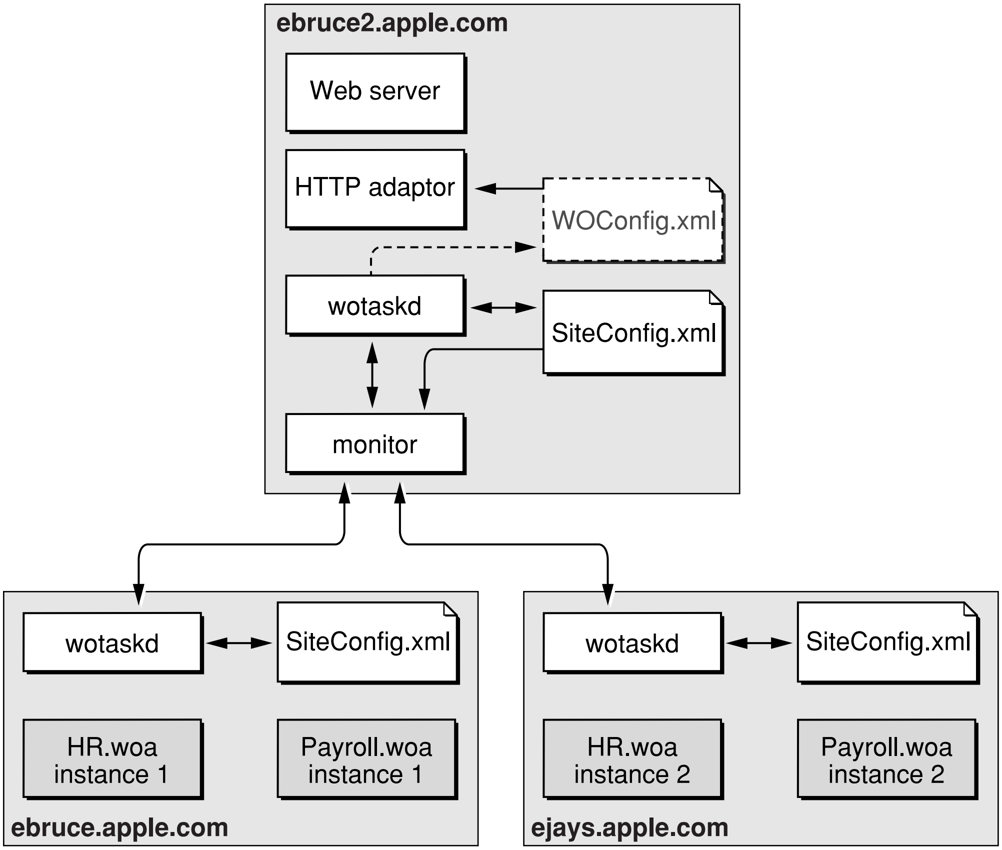
You use Monitor to configure your site. With it you configure hosts, applications, and application instances. You also define schedules for restarting instances and choose the algorithm used to balance the user load among the instances of an application. For details on the tasks that you can perform using Monitor, see "Deployment Tasks" (page 67).

The SiteConfig.xml file, which is maintained on each host's configuration directory by wotaskd, stores the configuration choices you make in Monitor. You should not modify the contents of this file directly. The user under which wotaskd runs must have read and write privileges to SiteConfig.xml and the user under which Monitor runs must have read privileges.

You can create another configuration file (the HTTP adaptor configuration file), which is called WOConfig.xml by default. This is the file the HTTP adaptor uses to obtain your site's configuration when you choose the configuration file method for the adaptor.

Figure 5-1 shows how the configuration files are distributed in a deployment with one Web server and two application hosts. For information on how to generate the HTTP adaptor configuration file, see "Creating the HTTP Adaptor Configuration File" (page 47).

Figure 5-1 WebObjects configuration-file distribution



Lifebeats

After adding an application to your site, you need to add instances of it as well. This allows the application's users connect to it. One of the main goals of WebObjects Deployment is to provide you with tools that help you deploy a resilient site. To accomplish that, wotaskd restarts application instances that become unresponsive.

A **lifebeat** is a status message that an application instance sends to a wotaskd process to keep it informed of the instance's status. There are four kinds of lifebeats:

- has started
- is alive
- will stop
- will crash

An application instance is configured by default to send lifebeats to a wotaskd process through TCP (Transmission Control Protocol) sockets. (A **socket** is a mechanism through which two processes communicate.) Instances can also send lifebeats using UDP (User Datagram Protocol) sockets. UDP sockets use fewer resources than TCP sockets.

The lifebeat mechanism is how the state of your site is constantly updated. Lifebeats are sent from a separate execution thread in a WebObjects application and do not interfere with, nor are they affected by, normal request processing.

By default, application instances try to send a lifebeat every 30 seconds. If a failure occurs (there is no wotaskd process listening on the `WOLifebeatDesinationPort`), the instance does the following:

1. Sends up to 10 TCP lifebeats until a response is received.

   After the tenth unanswered lifebeat, the instance goes to stage 2.

2. Sends UDP lifebeats (low-resource lifebeat mode) until a response is received.

   If the instance cannot create a UDP socket, it goes back to stage 1.

   When the instance receives an acknowledgement, it resumes sending TCP lifebeats.

The two-stage mechanism allows application instances to go into low-resource lifebeat mode if wotaskd isn't running when the instance starts.

wotaskd Processes

WebObjects Deployment uses wotaskd to manage the application instances running on your application hosts. Its main task is to start up instances when hosts are restarted. To accomplish this, wotaskd itself has to be restarted when the host starts up. This is done by configuring wotaskd as a service started when the computer boots. By default, a wotaskd process running on port `1085` is configured as a service on all supported platforms. The implementation of this feature is platform-specific. See "Starting WebObjects Services" (page 63) for details.

You need to run a wotaskd process on every machine that you want to use as an application host. These processes constantly receive lifebeats from the application instances they manage. Lifebeats communicate the instance's state and allow wostaskd to determine the state of the instances it oversees. A wotaskd process assumes that an application instance is dead if it does not receive a lifebeat within a certain period. For details, see "WOAssumeApplicationIsDeadMultiplier" (page 122).

When you restrict access to Monitor with a password, wotaskd processes running on hosts configured in Monitor are also protected: They do not respond to `http://<hostname>:<wotaskd-port>`, as described in "Confirming That wotaskd Is Active" (page 32).

To access the state information of a particular wotaskd process, you use Monitor's Host page. See "Viewing a Host's Configuration" (page 72) for more information.

Starting WebObjects Services

You use Monitor and wotaskd to build and troubleshoot your site. WebObjects provides scripts that allow you to control some of the behavior of Monitor and wotaskd. Specifically, you can determine whether they are started automatically when a machine starts up. That way, you'll have one less item to worry about if a machine goes down. On the other hand, if you prefer a more hands-on approach to site management, you can start and stop WebObjects services manually.

Starting WebObjects Services Automatically

The default installation of WebObjects Deployment in Mac OS X Server adds a startup script that automatically starts a wotaskd process during system startup. In addition, if the wotaskd process dies, it's automatically restarted.

You can configure a machine to keep wotaskd and Monitor processes active at all times. To accomplish that, you have to edit the `WebObjects` script file. In Mac OS X Server, it's located in the `/System/Library/StartupItems/WebObjects` directory; in Solaris, you'll find it in the `/etc/init.d` directory. For example, to always have a Monitor process running on port `8888` in your site-administration computer, you add the following line after the line that starts wotaskd in the `WebObjects` script file:

```
"$WOSERVICE" -appPath /System/Library/WebObjects/JavaApplications/
JavaMonitor.woa/JavaMonitor -WOPort 8888 >>/var/log/webobjects.log 2>&1 &
```

For more on the command-line arguments available, see "Command-Line Arguments" (page 118).

Starting Monitor Manually

To start Monitor, enter the following commands in your command shell editor:

```
cd ($NEXT_ROOT)/System/Library/WebObjects/JavaApplications/JavaMonitor.woa
./JavaMonitor
```

You should see output similar to that in Listing 5-1.

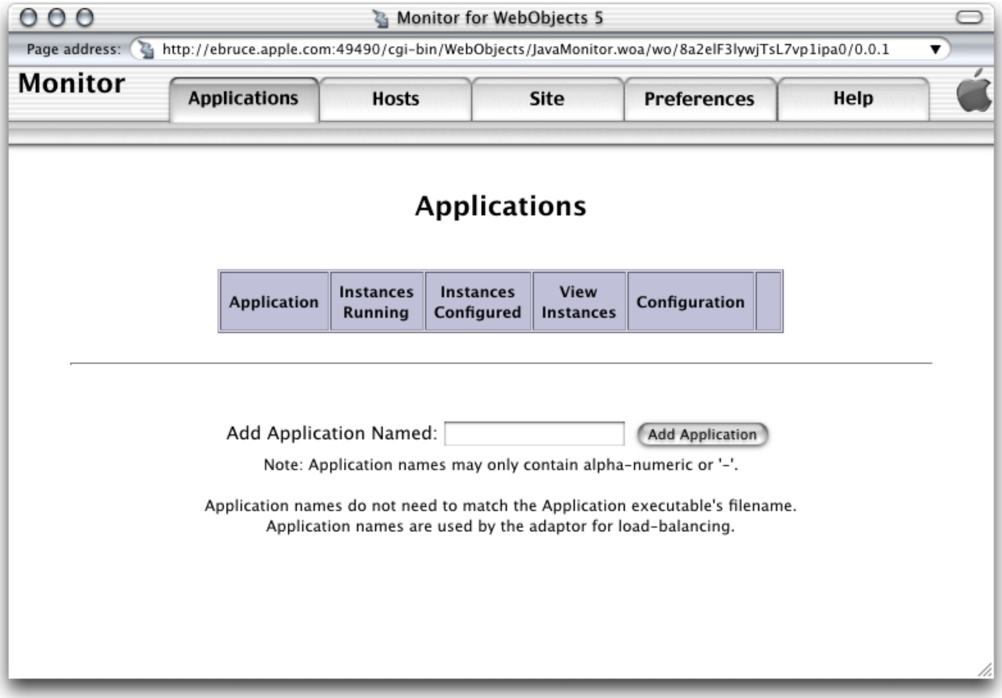
Listing 5-1 Starting Monitor

```
Reading MacOSClassPath.txt ...
Launching JavaMonitor.woa ...
...
Creating LifebeatThread now with: JavaMonitor 49490 ebruce.apple.com/
17.203.33.19 1085 30000
```

```
Opening application's URL in browser:
http://ebruce.apple.com:49490/cgi-bin/WebObjects/JavaMonitor
Waiting for requests...
```

A page like the one in Figure 5-2 should display in your Web browser. If your browser is not launched automatically, you can copy the URL from your shell and paste it into your browser's address field.

Figure 5-2 Monitor—empty Applications page



http://ebruce.apple.com:49490/cgi-bin/WebObjects/JavaMonitor.woa/wo/8a2elF3lywjTsL7vp1ipa0/0.0.1

Applications

Application	Instances Running	Instances Configured	View Instances	Configuration	

Add Application Named:

Note: Application names may only contain alpha-numeric or '-'.

Application names do not need to match the Application executable's filename.
Application names are used by the adaptor for load-balancing.

Controlling WebObjects Services in Solaris

The `WOServices` script, located in the `$NEXT_ROOT/Library/WebObjects/Executables` directory, provides you with four options to manage WebObjects services (such as wotaskd and Monitor) from the command line: start, stop, enable, and disable.

- `WOServices start`

  Starts WebObjects services.

- `WOServices stop`

  Stops WebObjects services.

- `WOServices enable [-altJVMPath` *path*`]`

  Starts WebObjects services every time the computer is restarted. The `altJVMPath` argument specifies a Java executable different from the one present during installation. It is used to set the `PATH` variable used when starting wotaskd. If you use the `altJVMPath` argument, make sure to specify a full path.

- `WOServices disable`

  Disables wotaskd processes from starting automatically during system startup.

Windows 2000

During installation on Windows 2000, wotaskd is configured to start automatically at boot time (in the Services control panel, it's listed as Apple WebObjects Task Daemon). If it doesn't start, check the Services control panel to ensure that wotaskd's startup mode is set to `Automatic`.

Monitor is also configured as a service (listed as Apple WebObjects Monitor in the Services control panel) and you can configure it to start automatically during the boot process by changing its startup mode to `Automatic`. Note that although this causes Monitor to start automatically, you have to manually start your Web browser and connect to the Monitor process manually (or by putting a shortcut to your Web browser in your Startup program group). The URL for Monitor can be verified by checking the Windows 2000 Event Viewer (choose Start > Programs > Administrative Tools > Event Viewer) and is similar to the following:

```
http://localhost:1027/cgi-bin/WebObjects.exe/JavaMonitor
```

If you don't have Monitor configured to start automatically, you can launch it by choosing Start > Programs > WebObjects > Monitor. In that case, it runs on port `56789`.

CHAPTER 6

Deployment Tasks

Now that you are acquainted with the deployment tools of WebObjects, you are ready to put that knowledge to work by deploying your applications.

This chapter explains how Monitor allows you to perform most of the tasks required to maintain a WebObjects application site with point-and-click ease. The sections below guide you through the process of adding and configuring hosts, applications, and instances. Also explained is the load-balancing mechanism used by the HTTP adaptor, which performs load balancing among the instances of an application (which can be distributed across more than one host) and how to set up email notifications so that you and your colleagues are notified when problems arise.

This chapter addresses the following subjects:

- "Setting Up Hosts" (page 68) explains how you add and configure hosts for application deployment. Note that this is different from configuring hosts in the HTTP adaptor.
- "Installing Applications" (page 74) shows you where to place application files and Web server resources on an application host before deployment.
- "Setting Up Applications" (page 75) details the ways you can customize a deployment. These include choosing a load-balancing algorithm and scheduling instances to restart at regular intervals.
- "Configuring Sites" (page 92) describes the site-wide properties available. These include configuring Monitor to use your SMTP (Simple Mail Transfer Protocol) server to send email notifications.

- "Setting Monitor Preferences" (page 94) shows you the Monitor-specific preferences you can use to tailor the tool's behavior.
- "Load Balancing" (page 97) explains how load balancing distributes user load among the running instances of an application in your site.
- "Deploying Multiple Sites" (page 98) shows how you can maintain multiple sites on one set of computers.

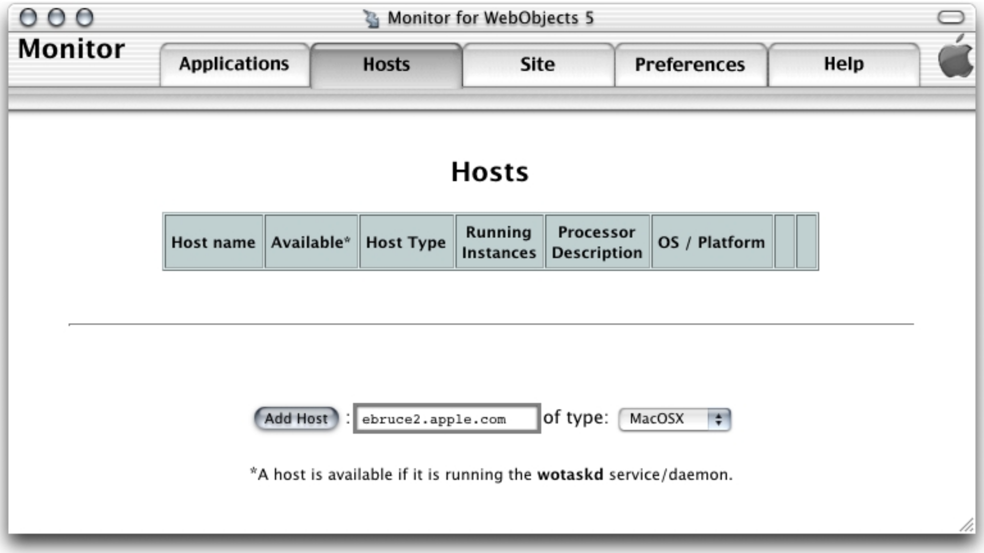
Setting Up Hosts

An application host is a machine that runs application instances. For Monitor to be able to identify a host, it has to be running a wotaskd process. See "Choosing What to Install" (page 29) and "Confirming That wotaskd Is Active" (page 32).

Adding a Host

Before you can deploy applications, you need to tell Monitor which hosts you want to use for deployment. (See "Load Balancing" (page 97) for additional information regarding load balancing and hosts added in Monitor.) Figure 6-1 shows Monitor's Hosts page, which you use to add and configure hosts.

Figure 6-1 The Hosts page



Hosts

Host name	Available*	Host Type	Running Instances	Processor Description	OS / Platform		

Add Host : ebruce2.apple.com of type: MacOSX

*A host is available if it is running the **wotaskd** service/daemon.

1. In Monitor, click the Hosts tab.
2. Enter the name or IP address of the host you want to add in the host text input field.

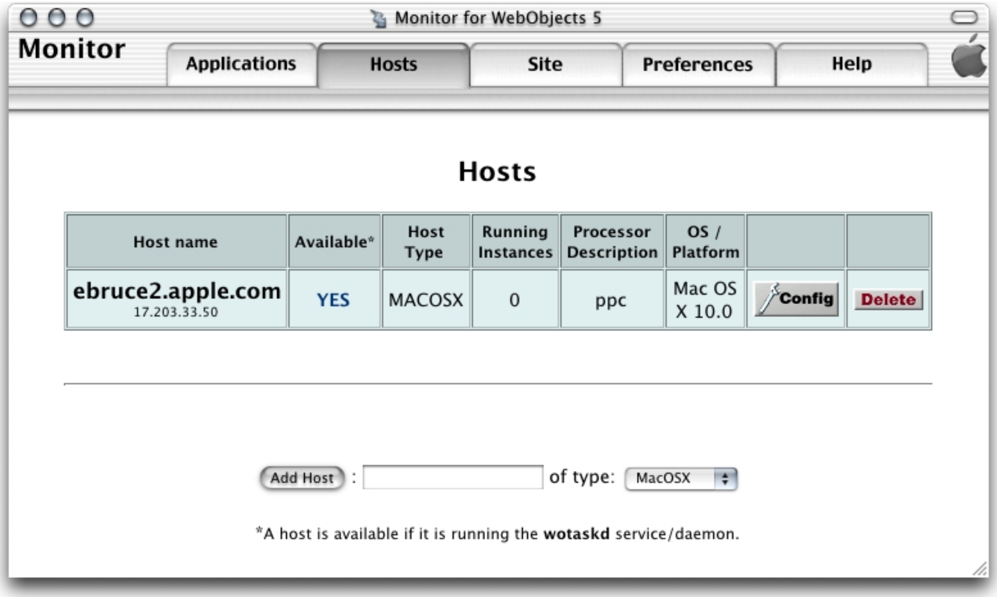
   The machine must be running a wotaskd process in the port that Monitor sends its lifebeats to.

   Avoid using a **loopback** address (a connection that does not go over the network), such as `localhost` or `127.0.0.1`. If you do, it must be the only application host in your site.

3. Chose the appropriate platform from the pop-up menu.
4. Click Add Host.

Figure 6-2 shows the Hosts page after a host has been added.

Figure 6-2 Newly added host in Monitor



Hosts

Host name	Available*	Host Type	Running Instances	Processor Description	OS / Platform		
ebruce2.apple.com 17.203.33.50	YES	MACOSX	0	ppc	Mac OS X 10.0	Config	Delete

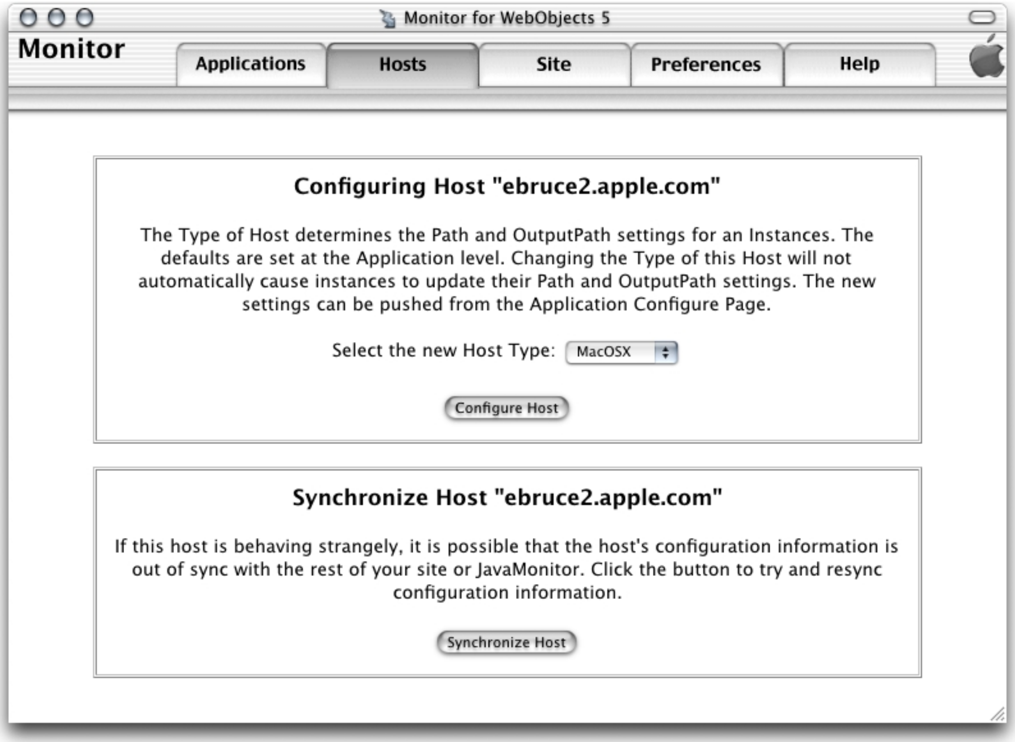
Add Host : of type: MacOSX

*A host is available if it is running the **wotaskd** service/daemon.

Configuring a Host

To change the configuration of an application host, click the Config button on the Hosts page. A page similar to the one in Figure 6-3 appears. It allows you to set the type of the host and to resynchronize configuration information if needed.

Figure 6-3 Host configuration page



Configuring Host "ebruce2.apple.com"

The Type of Host determines the Path and OutputPath settings for an Instances. The defaults are set at the Application level. Changing the Type of this Host will not automatically cause instances to update their Path and OutputPath settings. The new settings can be pushed from the Application Configure Page.

Select the new Host Type: MacOSX

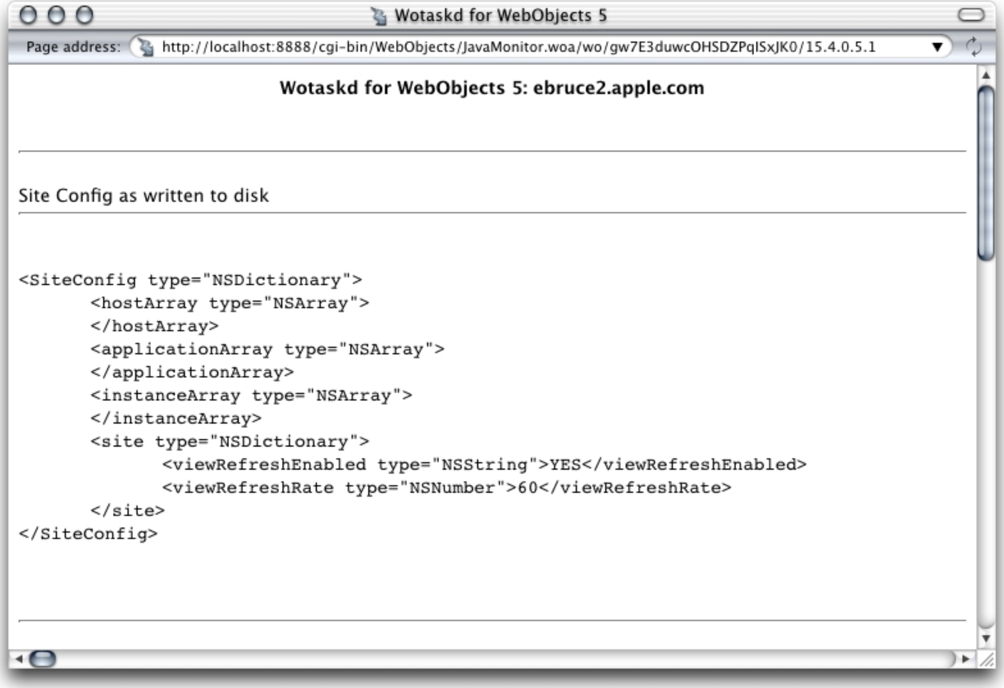
Synchronize Host "ebruce2.apple.com"

If this host is behaving strangely, it is possible that the host's configuration information is out of sync with the rest of your site or JavaMonitor. Click the button to try and resync configuration information.

Viewing a Host's Configuration

To display the configuration for a host, click YES, on the Hosts page. A page similar to the one in Figure 6-4 is displayed in a separate Web browser window.

Figure 6-4 Host configuration information page


Wotaskd for WebObjects 5
Page address:
http://localhost:8888/cgi-bin/WebObjects/JavaMonitor.woa/wo/gw7E3duwcOHSDZPqISxJK0/15.4.0.5.1

Wotaskd for WebObjects 5: ebruce2.apple.com

Site Config as written to disk

```
<SiteConfig type="NSDictionary">
        <hostArray type="NSArray">
        </hostArray>
        <applicationArray type="NSArray">
        </applicationArray>
        <instanceArray type="NSArray">
        </instanceArray>
        <site type="NSDictionary">
                <viewRefreshEnabled type="NSString">YES</viewRefreshEnabled>
                <viewRefreshRate type="NSNumber">60</viewRefreshRate>
        </site>
</SiteConfig>
```

This page displays the current state of the host in several sections.

The section visible in Figure 6-4 (page 72) shows the contents of the host's `SiteConfig.xml` file. For more information on the `SiteConfig.xml` file, see "Configuration Files" (page 59).

The next section shows the adaptor configuration sent to local HTTP adaptors, which lists all running application instances that wotaskd knows about (this includes Monitor processes).

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
    <application name="JavaMonitor">
        <instance id="-8888" port="8888" host="ebruce2.apple.com"/>
    </application>
</adaptor>
```

Note that the instance ID of the Monitor process is negative. When wotaskd receives a lifebeat from an application instance that has not been registered (added to your site through Monitor), it discloses the instance to the Web server with a negative ID number. This allows developers (internal users) to connect to instances through the HTTP adaptor for testing purposes. To address security concerns, external users can connect to instances with negative ID numbers only if they know the instance's port number.

To connect to a development instance through the Web server, the instance must run on the same computer that the Web server runs on, and the computer must be the `localhost`.

Next is the local adaptor configuration sent to remote HTTP adaptors. This section lists all instances that are active, registered (configured through Monitor), and available to external users.

```
<?xml version="1.0" encoding="ASCII"?>
<adaptor>
  <application name="Payroll" urlVersion="4">
    <instance id="1" port="2001" host="ebruce.apple.com"/>
  </application>
  <application name="HR" urlVersion="4">
    <instance id="1" port="2002" host="ebruce.apple.com"/>
  </application>
</adaptor>
```

The next section shows the contents of the HTTP adaptor configuration file. If you tell wotaskd to write the HTTP adaptor configuration file, it lists all the registered application instances on the site, whether they are running or not. (This file is identical across all the site's application hosts.) See "The HTTP Adaptor Configuration File" (page 44) for more information and an example of the file's contents.

The last section lists information on the wotaskd process's environment, including its port and multicast address.

```
The Configuration Directory is: /Library/WebObjects/Configuration/
Wotaskd is NOT writing WOConfig.xml to disk
The multicast address is: 239.128.14.2
This wotaskd is running on Port: 1085
Wotaskd is NOT responding to Multicast
WOAssumeApplicationIsDeadMultiplier is 4
The System Properties are: ...
```

Installing Applications

Before you can deploy applications on your site, you have to install them on the hosts on which you want to run instances of them. Most applications have files of two types:

- **application files,** which store the application's logic
- **Web server resources,** which store resources that can be shared among applications

The developer tool Project Builder (used to develop WebObjects applications) can be used to install an application on a host. With it, you can create an application bundle (a directory in the file system) with source code and resources the application needs to run. All the application files for the application can be included in the application bundle; however, doing so exposes the application's source to outside agents.

Instead, you should perform a split install, installing most of the application in a directory that is not accessible to the outside world. The nonsensitive resources can be placed on the Web server's `Document Root` directory.

To perform a split install of your application, navigate to your project's directory and execute the following command as `root` using your shell editor:

```
pbxbuild install -buildstyle WebServer DSTROOT=/
```

This places the application files in the `/Library/WebObjects/Applications` directory and the Web server resources in the `Document Root/WebObjects` directory.

Installing the Application Files

You can place application files in any directory of the application host.

Installing the Web Server Resources

You must place Web server resource files in the Web server `Document Root` directory of the application host. Make sure that you use the following organization:

```
<Web server Document Root>/
    WebObjects/
        AppName.woa/
            Contents/
                WebServerResources/
                    <resource files>
```

Setting Up Applications

To deploy an application on your site, it must be installed in the appropriate directories on the hosts that run instances of it. For more information, see "Installing Applications" (page 74).

Setting up an application in your site involves three main steps:

- **Adding the application,** which is described in "Adding an Application" (page 76).

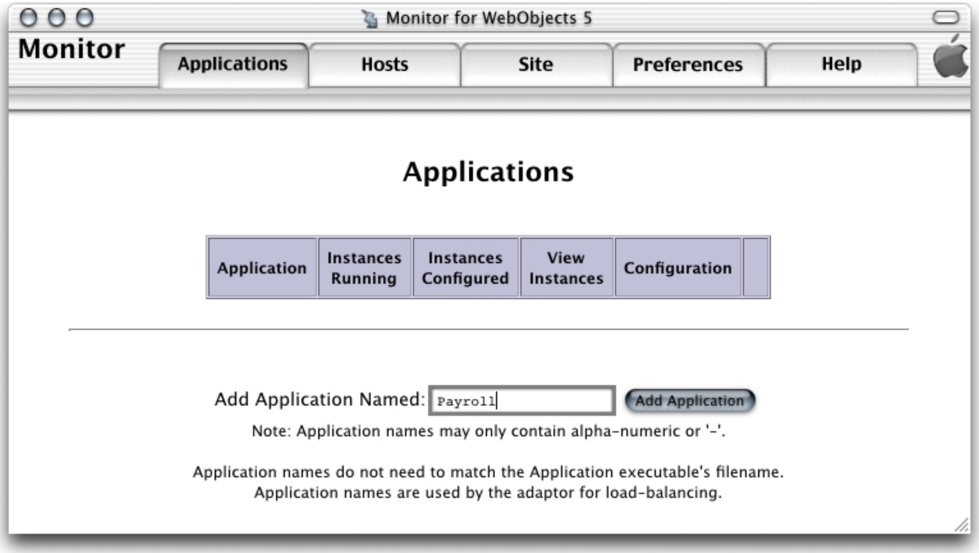
- **Configuring the application**, which includes the following subtasks:
  - □ defining a default configuration for new instances of the application
  - □ defining the recipients of email notifications
  - □ defining a schedule (only available for existing instances)
  - □ choosing a load-balancing algorithm

  These steps are detailed in "Configuring an Application" (page 77).
- **Adding application instances,** which is described in "Adding Application Instances" (page 84).

Adding an Application

You add applications to your site using Monitor's Applications page, shown in Figure 6-5.

Figure 6-5 Adding an application using Monitor's Applications page



Add Application Named: Payroll

Note: Application names may only contain alpha-numeric or '-'.

Application names do not need to match the Application executable's filename.
Application names are used by the adaptor for load-balancing.

Follow these steps to add an application:

1. Enter the application's name (without the extension) in the Add Application Named text field.
2. Click Add Application.

Configuring an Application

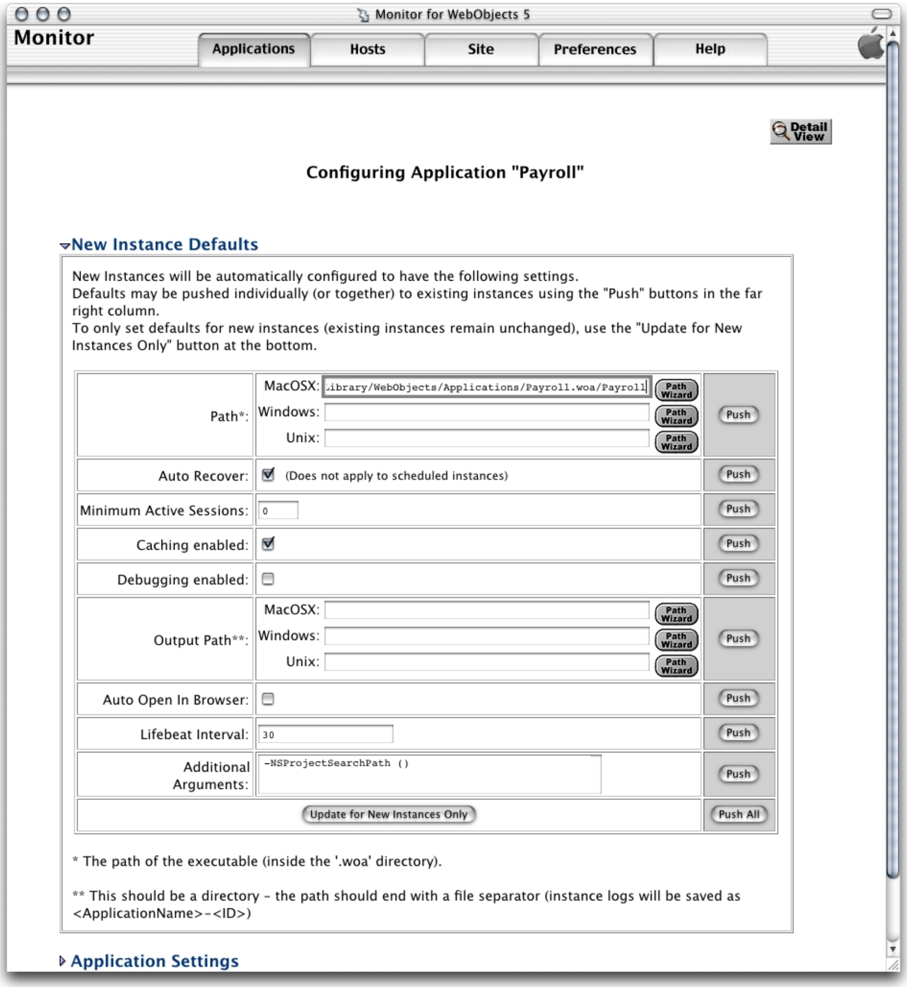
After you add an application, the application configuration page is displayed. This page has five major sections, which you can show and hide using the disclosure triangles:

- The New Instance Defaults section lets you set the default values for the instance settings for application instances you add afterward. See "New Instance Defaults" (page 78) for details.
- The Application Settings section contains properties that apply to all the instances of the application. For more, see "Application Settings" (page 80).
- The Scheduling section allows you to individually schedule instances to restart at specific intervals. See "Scheduling" (page 81) for details.
- The Email Notifications section is where you specify the list of email addresses you want email notifications to be sent to. For details, see "Email Notifications" (page 82).
- The Load Balancing and Adaptor Settings section lets you choose the algorithm that the HTTP adaptor uses to perform load balancing among the instances of the application. See "Load Balancing and Adaptor Settings" (page 83).

New Instance Defaults

Figure 6-6 shows the section of the application configuration page that allows you to set defaults for the application instances you create afterward and for current ones (which are updated after you restart them). For details on each of the properties shown on this page, see "Instance Settings" (page 113).

Figure 6-6 The New Instance Defaults section of the application configuration page



Configuring Application "Payroll"

▿New Instance Defaults

New Instances will be automatically configured to have the following settings.
Defaults may be pushed individually (or together) to existing instances using the "Push" buttons in the far right column.
To only set defaults for new instances (existing instances remain unchanged), use the "Update for New Instances Only" button at the bottom.

Path*:	MacOSX: Library/WebObjects/Applications/Payroll.woa/Payroll Windows: Unix:	Push
Auto Recover:	☑ (Does not apply to scheduled instances)	Push
Minimum Active Sessions:	0	Push
Caching enabled:	☑	Push
Debugging enabled:	☐	Push
Output Path**:	MacOSX: Windows: Unix:	Push
Auto Open In Browser:	☐	Push
Lifebeat Interval:	30	Push
Additional Arguments:	-NSProjectSearchPath ()	Push
	Update for New Instances Only	Push All

* The path of the executable (inside the '.woa' directory).

** This should be a directory - the path should end with a file separator (instance logs will be saved as <ApplicationName>-<ID>)

▹ Application Settings

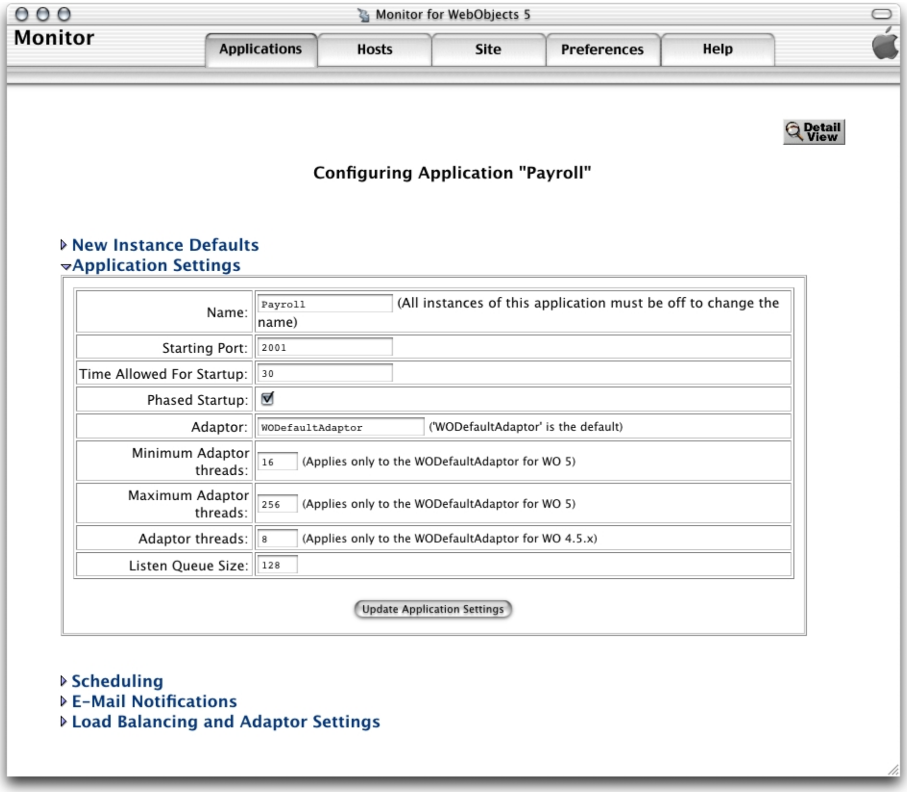
Here's an explanation of the buttons you see on the page:

- **Push** updates the value of the property for new and configured (registered) instances of the application. The changes take effect after the instances are restarted.
- **Push All** updates the value of the all the properties for new and configured instances of the application. As with Push, the changes become effective when the instances are restarted.
- **Update for New Instances Only** sets the defaults to be used when you create new instances of the application. The properties of existing instances are not changed.
- **Path Wizard** opens a tool that allows you to navigate through a host's file system.

Application Settings

Figure 6-7 shows the Application Settings section of the application configuration page. In it, you define application settings that apply to all the instances of the application. For details of the properties shown in this section, see "Application Settings" (page 112).

Figure 6-7 The Application Settings section of the Applications page



Configuring Application "Payroll"

- New Instance Defaults
- Application Settings

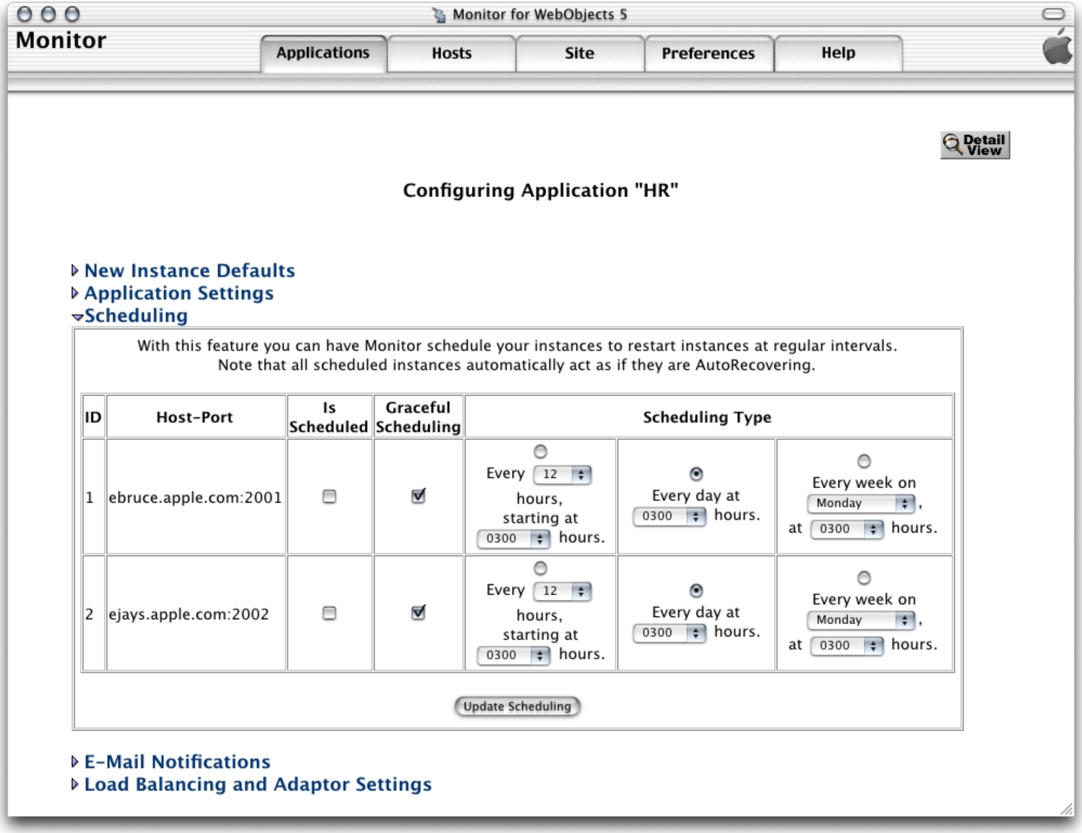
Name:	Payroll (All instances of this application must be off to change the name)
Starting Port:	2001
Time Allowed For Startup:	30
Phased Startup:	☑
Adaptor:	WODefaultAdaptor ('WODefaultAdaptor' is the default)
Minimum Adaptor threads:	16 (Applies only to the WODefaultAdaptor for WO 5)
Maximum Adaptor threads:	256 (Applies only to the WODefaultAdaptor for WO 5)
Adaptor threads:	8 (Applies only to the WODefaultAdaptor for WO 4.5.x)
Listen Queue Size:	128

- Scheduling
- E-Mail Notifications
- Load Balancing and Adaptor Settings

Scheduling

Figure 6-8 shows the Scheduling section of the application configuration page. After you add instances of an application, you can schedule them individually here. For details, see "Scheduling Settings" (page 117).

Figure 6-8 The Scheduling section of the application configuration page



Configuring Application "HR"

▸ New Instance Defaults
▸ Application Settings
▾ Scheduling

With this feature you can have Monitor schedule your instances to restart instances at regular intervals.
Note that all scheduled instances automatically act as if they are AutoRecovering.

ID	Host-Port	Is Scheduled	Graceful Scheduling	Scheduling Type		
1	ebruce.apple.com:2001	□	☑	○ Every 12 hours, starting at 0300 hours.	● Every day at 0300 hours.	○ Every week on Monday, at 0300 hours.
2	ejays.apple.com:2002	□	☑	○ Every 12 hours, starting at 0300 hours.	● Every day at 0300 hours.	○ Every week on Monday, at 0300 hours.

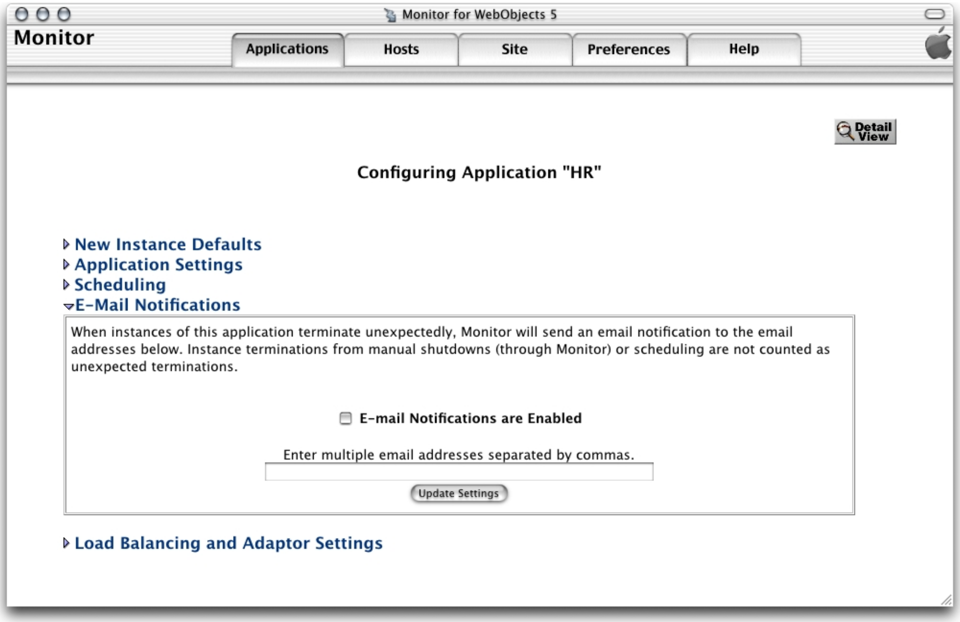
▸ E-Mail Notifications
▸ Load Balancing and Adaptor Settings

Email Notifications

Note: Before you can configure email notifications, you have to tell Monitor which SMTP server to use. See "Configuring Sites" (page 92).

Figure 6-9 shows the Email Notification Settings section of the application configuration page. In it you can enter the email addresses of people that are to be notified when instances of the application terminate unexpectedly.

Figure 6-9 The Email Notifications section of the Application Configuration page



Configuring Application "HR"

When instances of this application terminate unexpectedly, Monitor will send an email notification to the email addresses below. Instance terminations from manual shutdowns (through Monitor) or scheduling are not counted as unexpected terminations.

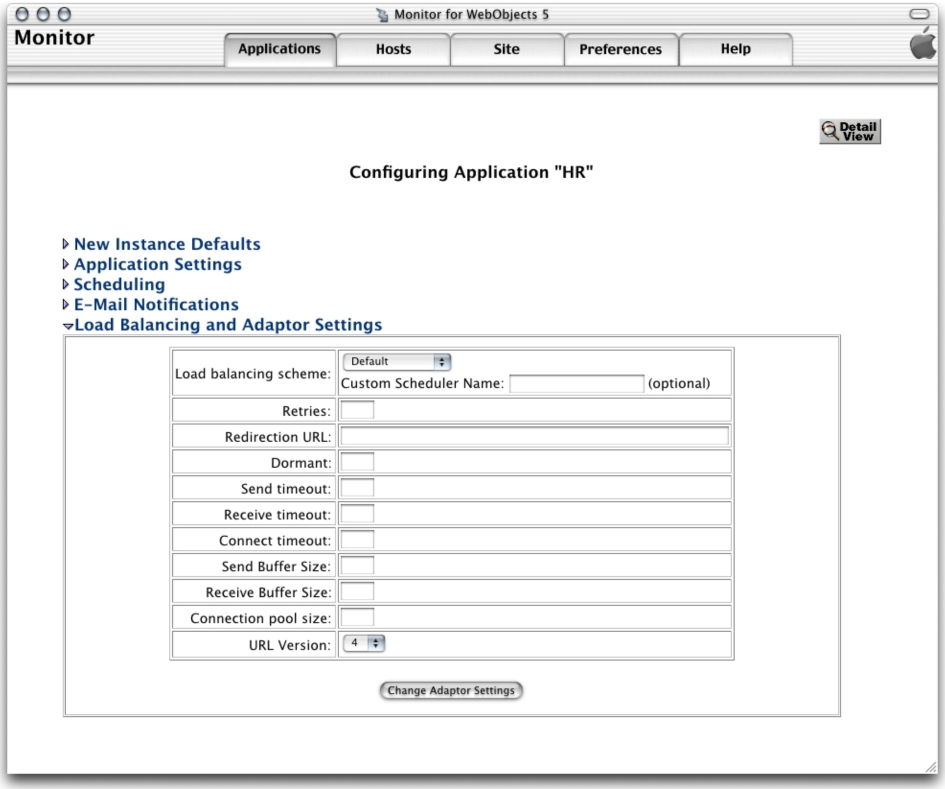
□ E-mail Notifications are Enabled

Enter multiple email addresses separated by commas.

Load Balancing and Adaptor Settings

Figure 6-10 shows the Load Balancing and Adaptor Settings section of the application configuration page. This is where you enter values for the HTTP adaptor's configuration properties, including the load-balancing algorithm the adaptor will use to balance user load among the application's instances. These settings override the values entered in the HTTP Adaptor Settings section of the Site page. For information on the properties you can set, see "Load Balancing and Adaptor Settings" (page 115).

Figure 6-10 The Load Balancing and Adaptor Settings section of the application configuration page



Configuring Application "HR"

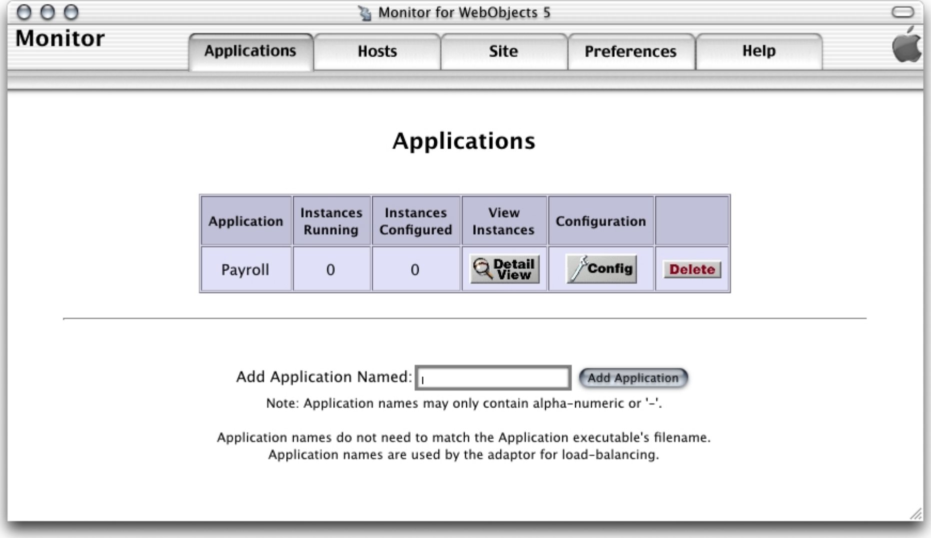
Load balancing scheme:	Default Custom Scheduler Name: (optional)
Retries:	
Redirection URL:	
Dormant:	
Send timeout:	
Receive timeout:	
Connect timeout:	
Send Buffer Size:	
Receive Buffer Size:	
Connection pool size:	
URL Version:	4

Adding Application Instances

After you have configured an application in Monitor, you can create application instances with ease.

1. In Monitor, click the Applications tab. The Applications page is displayed, as shown in Figure 6-11.

Figure 6-11 The Applications page with one application



Applications

Application	Instances Running	Instances Configured	View Instances	Configuration	
Payroll	0	0	Detail View	Config	Delete

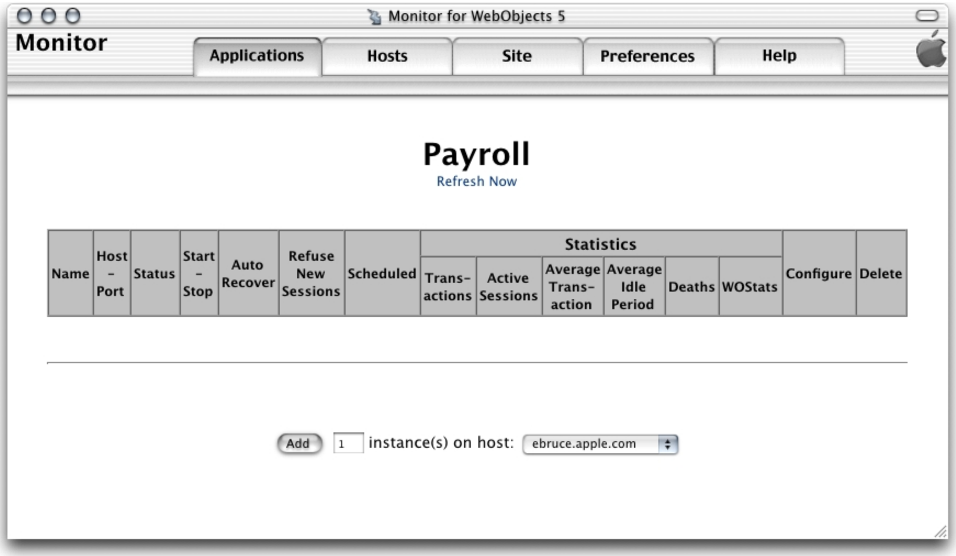
Add Application Named:

Note: Application names may only contain alpha-numeric or '-'.

Application names do not need to match the Application executable's filename.
Application names are used by the adaptor for load-balancing.

2. Click the Detail View button under View Instances. The application detail page is displayed, as shown in Figure 6-12.

Figure 6-12 The application detail page



Payroll

Refresh Now

Name	Host - Port	Status	Start - Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics						Configure	Delete
							Transactions	Active Sessions	Average Transaction	Average Idle Period	Deaths	WOStats		

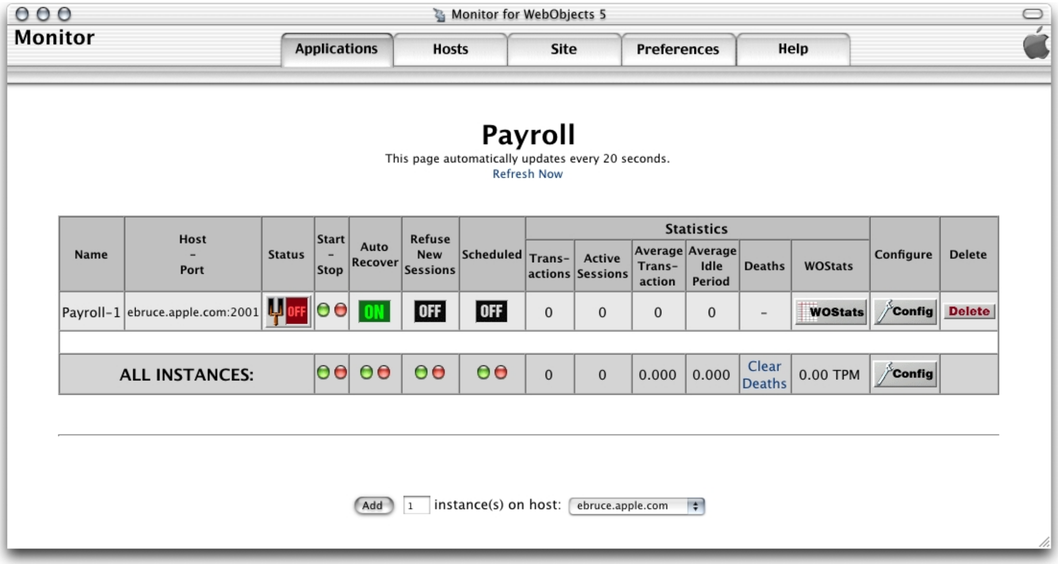
Add 1 instance(s) on host: ebruce.apple.com

3. Enter the number of instances you want to add in the text input field.
4. Choose the application host you want the instances to run on from the pop-up menu.

   The application must be installed on the host you choose; otherwise, an error message is displayed when you try to start the instance. See "Installing Applications" (page 74) for details.

5. Click Add. Your Web browser displays a page like the one in Figure 6-13.

Figure 6-13 The application detail page after an instance has been added



Payroll

This page automatically updates every 20 seconds.
Refresh Now

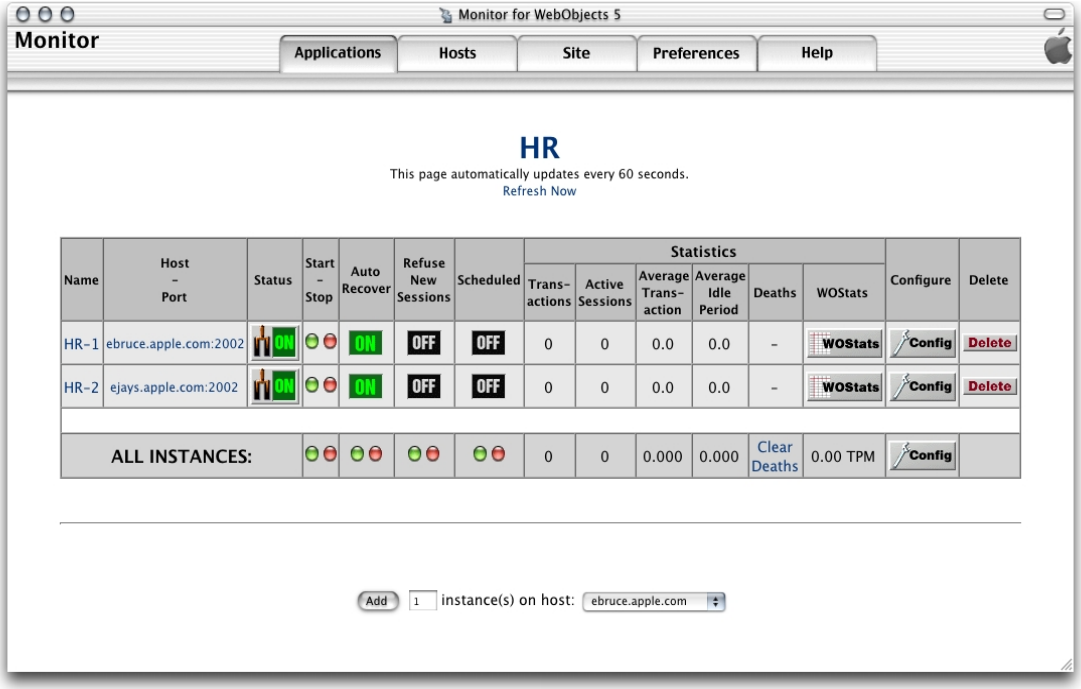
Name	Host - Port	Status	Start - Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics						Configure	Delete
							Trans-actions	Active Sessions	Average Trans-action	Average Idle Period	Deaths	WOStats		
Payroll-1	ebruce.apple.com:2001	OFF	● ●	ON	OFF	OFF	0	0	0	0	-	WOStats	Config	Delete
ALL INSTANCES:			● ●	● ●	● ●	● ●	0	0	0.000	0.000	Clear Deaths	0.00 TPM	Config	

Add [1] instance(s) on host: ebruce.apple.com

The Status column indicates whether the instance is on or off. The first time the page is displayed, the newly added instances are off. After a moment (or if you click Refresh Now), and if Auto Recover is enabled for the instance, the page refreshes, showing that the instance is active. (You can change the length of the interval between the automatic updates of the application detail page; see "Setting Monitor Preferences" (page 94) for details.)

Figure 6-14 shows the application detail page of the HR application, with two instances configured.

Figure 6-14 The Application Detail page with two instances added



HR

This page automatically updates every 60 seconds.
Refresh Now

Name	Host - Port	Status	Start - Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics						Configure	Delete
							Trans-actions	Active Sessions	Average Trans-action	Average Idle Period	Deaths	WOStats		
HR-1	ebruce.apple.com:2002	ON	🟢🔴	ON	OFF	OFF	0	0	0.0	0.0	-	WOStats	Config	Delete
HR-2	ejays.apple.com:2002	ON	🟢🔴	ON	OFF	OFF	0	0	0.0	0.0	-	WOStats	Config	Delete
ALL INSTANCES:			🟢🔴	🟢🔴	🟢🔴	🟢🔴	0	0	0.000	0.000	Clear Deaths	0.00 TPM	Config	

Add 1 instance(s) on host: ebruce.apple.com

The following list describes the instance configuration information that appears in the application detail page:

- **Page heading:** A link to the application through the HTTP adaptor. When you click it, the adaptor uses load balancing to determine which of the application's instances receives the request. Then, your Web browser displays a new window showing your application's entry page. For this to work, the HTTP adaptor URL has to be set ("Configuring Sites" (page 92) shows you how to do this).

- **Name:** A link to the instance through the HTTP adaptor. When you click it, the request goes through the adaptor, but it's not load balanced. The URL is derived in the same way the URL of the page heading is derived, but with the addition of the instance number. Such a URL looks similar to the following:

  ```
  http://ebruce2.apple.com/cgi-bin/WebObjects/HR.woa/1
  ```

- **Host-Port:** A direct link to the instance; does not go through the HTTP adaptor.
- **Status:** Tells whether the instance is on, off, starting, or stopping.
- **Start-Stop:** Click the green button to turn the instance on, or the red button to turn it off.
- **Auto Recover:** Click to toggle between ON and OFF. This is available only if the instance is not scheduled. See "Auto Recover" (page 114).
- **Refuse New Sessions:** Click to toggle between ON and OFF. When ON, the instance does not accept new users. This is available only if the instance is not scheduled.
- **Scheduled:** Click to toggle between ON and OFF. When ON, the schedule defined for the instance is used.
- **Configure:** Click the Config button to go to the instance configuration page for the instance.
- **Delete:** Click the Delete button to delete the instance. You'll see a confirmation page before the deletion takes place.

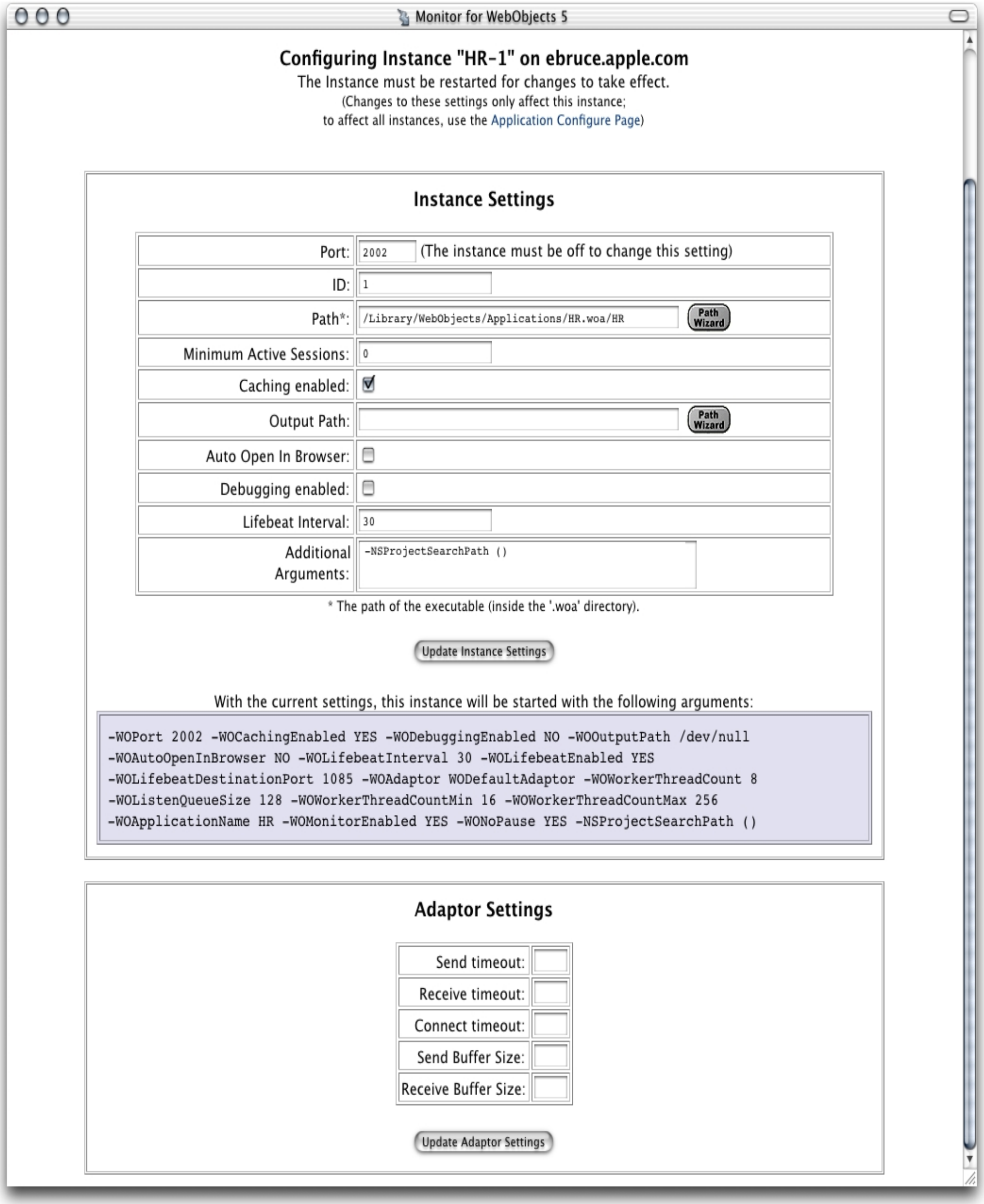
For an explanation of the columns under Statistics, see "The Application Detail Page" (page 102).

The row with the caption ALL INSTANCES contains buttons that perform some of the functions listed above on all the instances of the application. Clicking Config displays the application configuration page.

Configuring Instances

After you have added an instance, you can change its configuration in the instance configuration page, shown in Figure 6-15. You can access this page through the instance's Config button in the application detail page. It contains two sections: Instance Settings and Adaptor Settings.

Figure 6-15 Instance configuration page



Configuring Instance "HR-1" on ebruce.apple.com

The Instance must be restarted for changes to take effect.

(Changes to these settings only affect this instance;
to affect all instances, use the Application Configure Page)

Instance Settings

Port:	2002 (The instance must be off to change this setting)
ID:	1
Path*:	/Library/WebObjects/Applications/HR.woa/HR
Minimum Active Sessions:	0
Caching enabled:	☑
Output Path:	
Auto Open In Browser:	□
Debugging enabled:	□
Lifebeat Interval:	30
Additional Arguments:	-NSProjectSearchPath ()

* The path of the executable (inside the '.woa' directory).

With the current settings, this instance will be started with the following arguments:

```
-WOPort 2002 -WOCachingEnabled YES -WODebuggingEnabled NO -WOOutputPath /dev/null
-WOAutoOpenInBrowser NO -WOLifebeatInterval 30 -WOLifebeatEnabled YES
-WOLifebeatDestinationPort 1085 -WOAdaptor WODefaultAdaptor -WOWorkerThreadCount 8
-WOListenQueueSize 128 -WOWorkerThreadCountMin 16 -WOWorkerThreadCountMax 256
-WOApplicationName HR -WOMonitorEnabled YES -WONoPause YES -NSProjectSearchPath ()
```

Adaptor Settings

Send timeout:	
Receive timeout:	
Connect timeout:	
Send Buffer Size:	
Receive Buffer Size:	

Instance Settings

This section is very similar to the New Instance Defaults section of the application configuration page. It has two additional properties: ID and Port, which can only be changed after an instance has been added. For details, see "ID" (page 114) and "Port" (page 115).

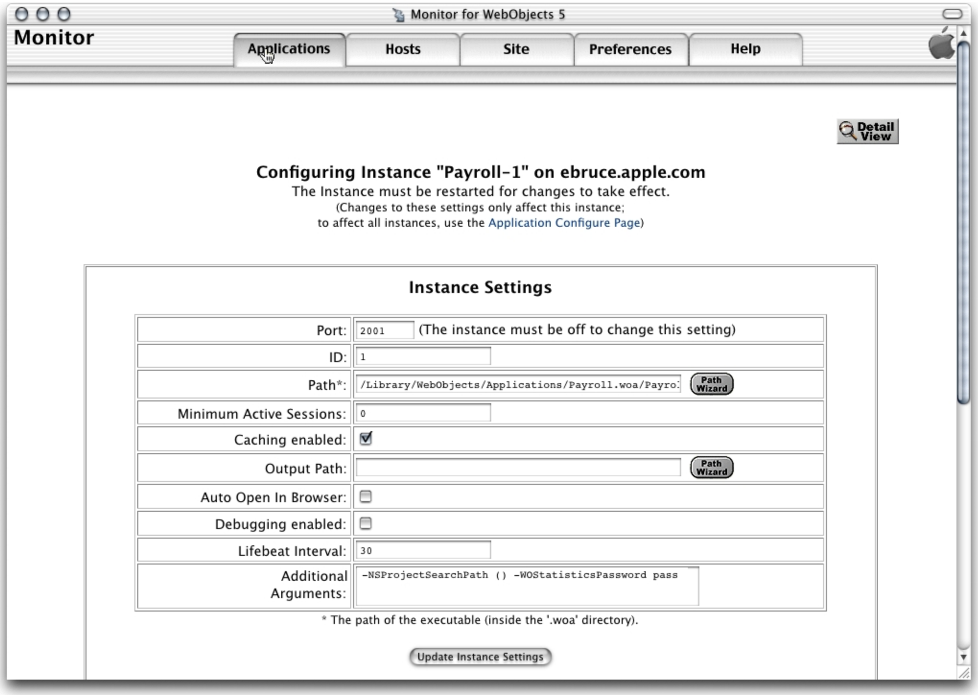
Adaptor Settings

In this section you can change a subset of the properties available in the Load Balancing and Adaptor Settings section of the application configuration page. For details, see "Load Balancing and Adaptor Settings" (page 115).

Setting a Password for the Instance Statistics Page

For each instance of your application, there's a statistics page that displays information such as its running time and memory usage. See "The Instance Statistics Page" (page 104) for more information on this page. If you want to prevent outside agents from gaining access to the instance statistics page, you can set a password in the Instance Settings section of the instance configuration page. Add the following to the Additional Arguments property: `-WOStatisticsPassword` *password*. Figure 6-16 shows an example where the `WOStatisticsPassword` argument has been added to the Additional Arguments field.

Figure 6-16 Setting a password for an instance's statistics page



Configuring Instance "Payroll-1" on ebruce.apple.com

The Instance must be restarted for changes to take effect.

(Changes to these settings only affect this instance;
to affect all instances, use the Application Configure Page)

Instance Settings

Port:	2001 (The instance must be off to change this setting)
ID:	1
Path*:	/Library/WebObjects/Applications/Payroll.woa/Payro:
Minimum Active Sessions:	0
Caching enabled:	☑
Output Path:	
Auto Open In Browser:	□
Debugging enabled:	□
Lifebeat Interval:	30
Additional Arguments:	-NSProjectSearchPath () -WOStatisticsPassword pass

* The path of the executable (inside the '.woa' directory).

Configuring Sites

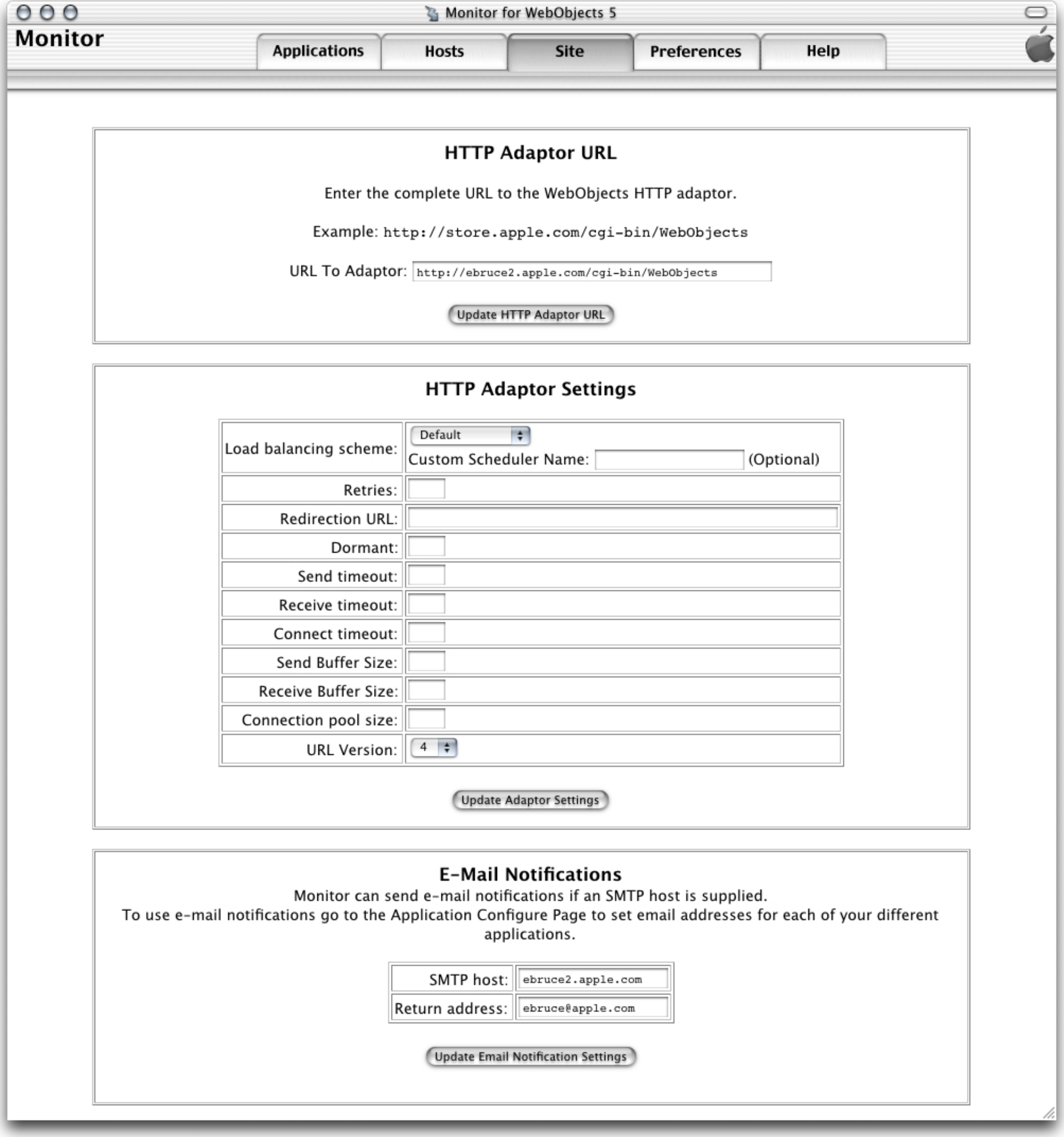
When you click the Site tab in Monitor, the site configuration page is displayed. It contains three sections:

- **HTTP Adaptor URL:** This is where you tell Monitor how to compose an application's URL, which used in the application detail page to connect you to instances of your application.

  To set the URL for your application, enter the URL in the URL to Adaptor field.
- **HTTP Adaptor Settings:** This is where you set default HTTP adaptor settings for all your deployed applications. They can be overridden by each application. For more information, see "Load Balancing and Adaptor Settings" (page 83) and "Load Balancing and Adaptor Settings" (page 115).
- **Email Notifications:** Here's where you specify the SMTP host and the return address that Monitor uses for email notifications.

Figure 6-17 shows the site configuration page:

Figure 6-17 The Site Configuration page



HTTP Adaptor URL

Enter the complete URL to the WebObjects HTTP adaptor.

Example: `http://store.apple.com/cgi-bin/WebObjects`

URL To Adaptor: `http://ebruce2.apple.com/cgi-bin/WebObjects`

HTTP Adaptor Settings

Load balancing scheme:	Default Custom Scheduler Name: (Optional)
Retries:	
Redirection URL:	
Dormant:	
Send timeout:	
Receive timeout:	
Connect timeout:	
Send Buffer Size:	
Receive Buffer Size:	
Connection pool size:	
URL Version:	4

E-Mail Notifications

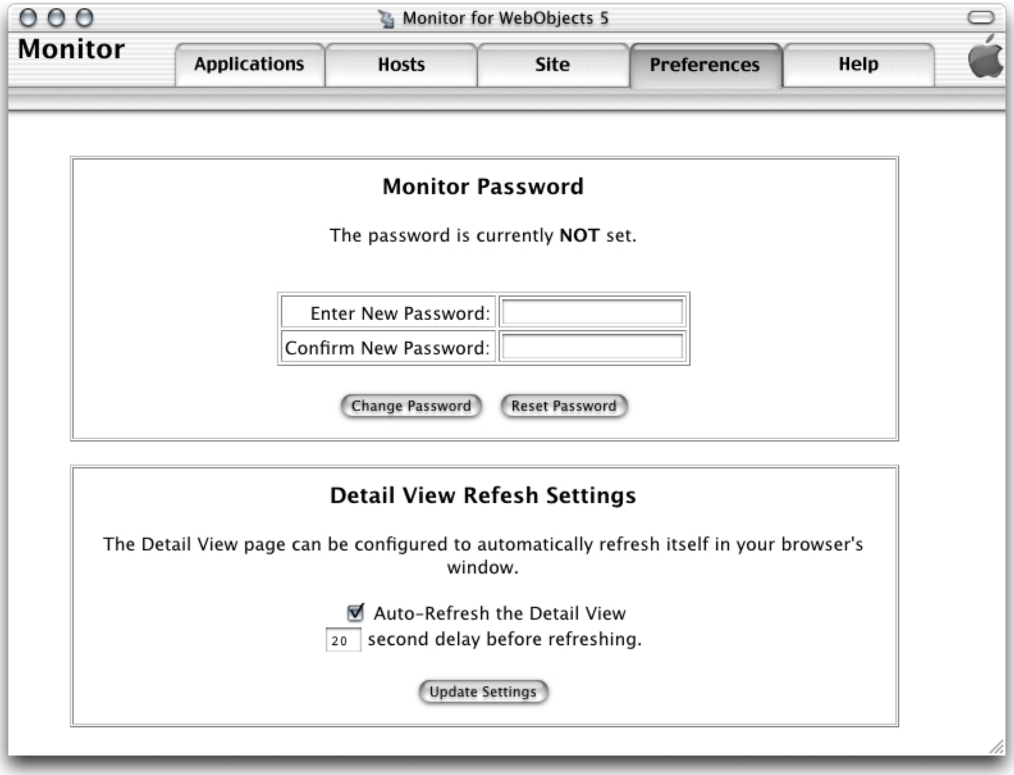
Monitor can send e-mail notifications if an SMTP host is supplied.
To use e-mail notifications go to the Application Configure Page to set email addresses for each of your different applications.

SMTP host:	ebruce2.apple.com
Return address:	ebruce@apple.com

Setting Monitor Preferences

When you click the Preferences tab in Monitor, the page in Figure 6-18 is displayed. It contains two sections: Monitor Password and Detail View Refresh Settings.

Figure 6-18 The Preferences page of Monitor



Monitor Password

The password is currently **NOT** set.

Enter New Password:	
Confirm New Password:	

Detail View Refesh Settings

The Detail View page can be configured to automatically refresh itself in your browser's window.

☑ Auto-Refresh the Detail View

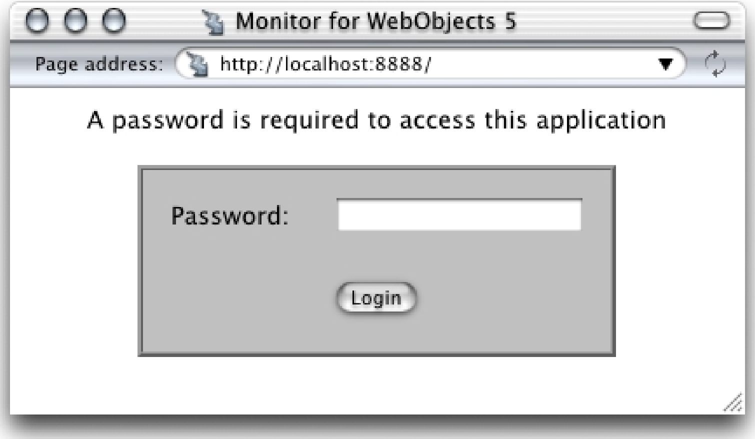
20 second delay before refreshing.

Monitor Password

You can restrict access to Monitor by requiring its users to enter a password before they can use it. When a site is protected this way, the site's wotaskd processes are also protected; that is, you cannot directly obtain a wotaskd process's information by connecting to its port, as described in "Confirming That wotaskd Is Active" (page 32).

Figure 6-19 shows the login page that Monitor displays after you password-protect your site.

Figure 6-19 Login page displayed by Monitor on a password-protected site

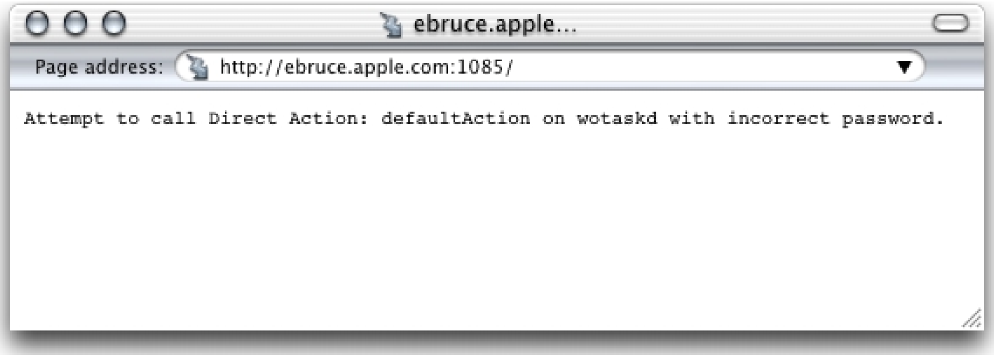


A password is required to access this application

Password:

When you try to view the configuration of an application host on a site that you've password-protected by connecting to the appropriate wotaskd process's port, you'll see a page similar to the one shown in Figure 6-20 (the page's content varies according to your deployment platform).

Figure 6-20 Page returned by wotaskd when the site is password-protected



```
Attempt to call Direct Action: defaultAction on wotaskd with incorrect password.
```

On password-protected sites, you'll have to use Monitor to view an application host's configuration.

Detail View Refresh Settings

This section allows you to tell Monitor if you want it to refresh the application detail page and how often to do it.

Load Balancing

Note: Load balancing occurs only with the hosts that the HTTP adaptor knows about. Adding a host in Monitor is not enough. For more information on how to configure hosts in the HTTP adaptor, see "State Discovery" (page 39).

Load balancing is a mechanism by which user-load is spread out among the instances of an application; these instances can be running on different hosts. Load balancing ensures that your site's hardware resources are used efficiently and with the highest level of performance. The default load-balancing algorithm used is Random. The following list describes how user load is distributed under each of the provided algorithms:

- **Random** assigns a user to an arbitrarily-chosen instance.
- **Round-Robin** assigns users among instances sequentially.
- **Load Average** balances load by distributing users evenly among instances.

You can choose a load-balancing algorithm at two levels:

- **Site level:** You set a site-wide load-balancing algorithm in the HTTP Adaptor Settings section of the Site page. From then on, the applications in your site will use that load-balancing algorithm.
- **Application level:** You can override the site-wide load-balancing algorithm in the Load Balancing and Adaptor Settings section of the application configuration page.

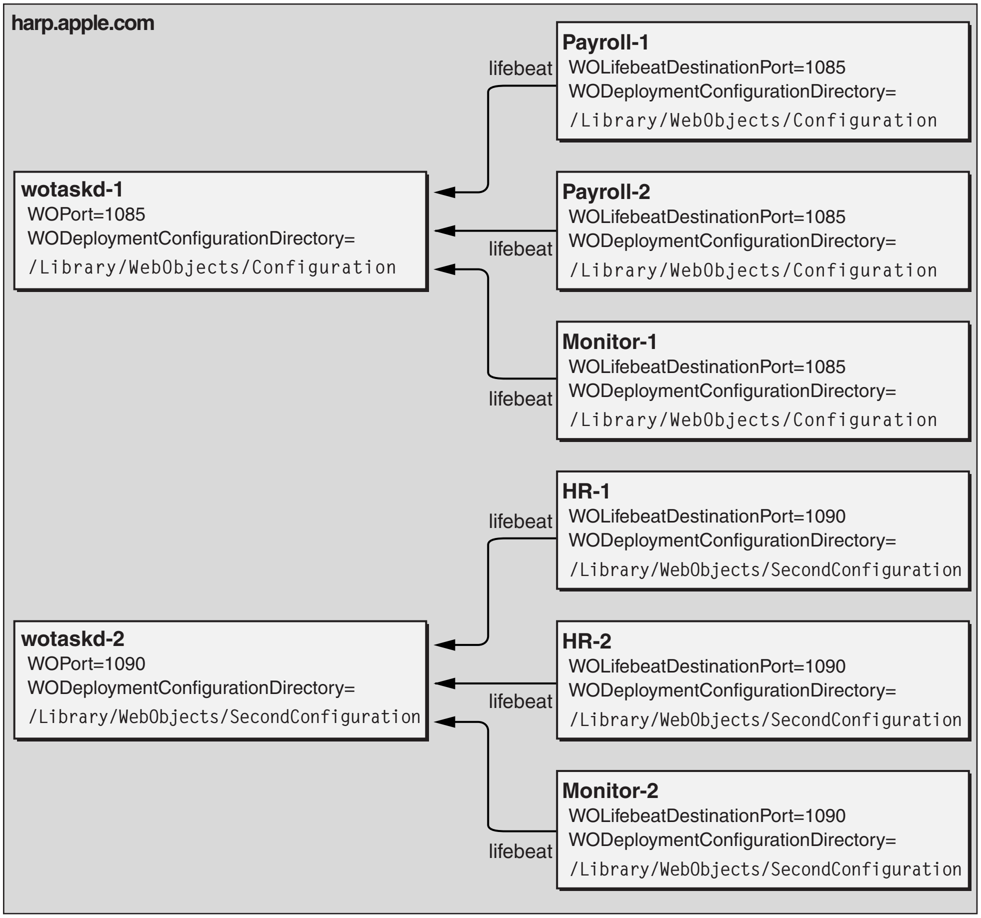
Deploying Multiple Sites

You can deploy and configure separate sites on a set of computers by running multiple Web servers, each with its own adaptor. Such a deployment requires a separate group of wotaskd processes running on the same port. You also need an additional Monitor process to configure each site.

The default installation of the WebObjects Deployment package provides you with one site—one wotaskd process per host, running on port `1085`. To create a second site, using the same hardware, you'll have to add an additional wotaskd process to each of the hosts you want to use.

What separates the environments from each other are the `WOPort` and `WOLifebeatDestinationPort` settings of each wotaskd process, and the configuration directory used for each site. The application instances send their lifebeats to their WOLifebeatDestinationPort, while wotaskd processes listen for them in their `WOPort`. Figure 6-21 illustrates two sites on one host.

Figure 6-21 Multiple application environments on one computer


harp.apple.com
Payroll-1
WOLifebeatDestinationPort=1085
WODeploymentConfigurationDirectory=
/Library/WebObjects/Configuration
lifebeat
wotaskd-1
WOPort=1085
WODeploymentConfigurationDirectory=
/Library/WebObjects/Configuration
Payroll-2
WOLifebeatDestinationPort=1085
WODeploymentConfigurationDirectory=
/Library/WebObjects/Configuration
lifebeat
Monitor-1
WOLifebeatDestinationPort=1085
WODeploymentConfigurationDirectory=
/Library/WebObjects/Configuration
lifebeat
HR-1
WOLifebeatDestinationPort=1090
WODeploymentConfigurationDirectory=
/Library/WebObjects/SecondConfiguration
lifebeat
wotaskd-2
WOPort=1090
WODeploymentConfigurationDirectory=
/Library/WebObjects/SecondConfiguration
HR-2
WOLifebeatDestinationPort=1090
WODeploymentConfigurationDirectory=
/Library/WebObjects/SecondConfiguration
lifebeat
Monitor-2
WOLifebeatDestinationPort=1090
WODeploymentConfigurationDirectory=
/Library/WebObjects/SecondConfiguration
lifebeat

Because Monitor is not started by a wotaskd process, its WOLifebeatDestinationPort argument needs to be set to match wotaskd's.

For details on how to set the different command-line argument values required when starting wotaskd and Monitor processes for separate application environments, see "WOPort" (page 120), "WOLifebeatDestinationPort" (page 119), and "WODeploymentConfigurationDirectory" (page 121).

CHAPTER 7

Application Administration

After deploying applications, you should monitor their performance to find out, among other things, if you need to add instances to an application to improve response times. "Monitoring Activity" (page 101) shows you the kind of performance data you can collect about an application.

There are several steps you can take to improve your site's performance; most of them have been explained in the preceding chapters. In the section "Improving Performance" (page 109) you'll find list of recommended measures that will help optimize your site's operation.

Monitoring Activity

There are several ways to obtain information about the applications deployed on your site. You can

- use Monitor
- analyze logs from application instances and adaptors
- view instance statistics (WOStats) pages

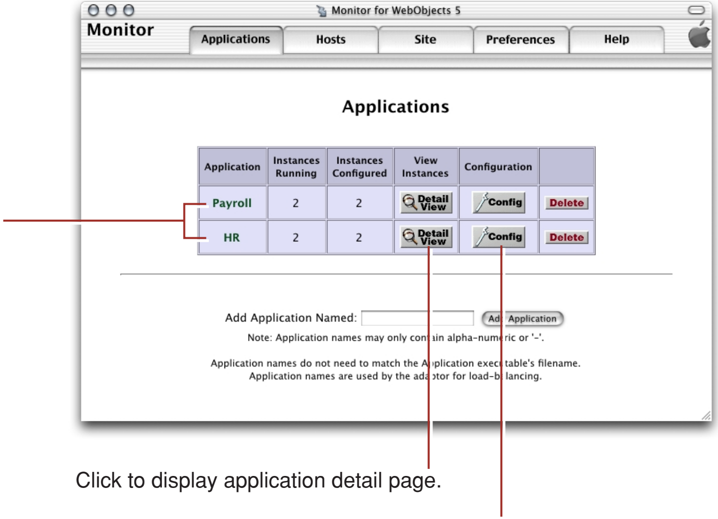
Monitoring Application Performance

Monitor's Applications page gives you an overall view of a site. Figure 7-1 illustrates the kind of information you can access through the Applications page:

- configured applications

- number of configured instances per application
- number of running instances per application

Figure 7-1 The Applications page



Applications

Application	Instances Running	Instances Configured	View Instances	Configuration	
Payroll	2	2	Detail View	Config	Delete
HR	2	2	Detail View	Config	Delete

Add Application Named: Add Application

Note: Application names may only contain alpha-numeric or '-'.

Application names do not need to match the Application executable's filename.
Application names are used by the adaptor for load-balancing.

Click to open application in new browser window.

Click to display application detail page.

Click to display application configuration page.

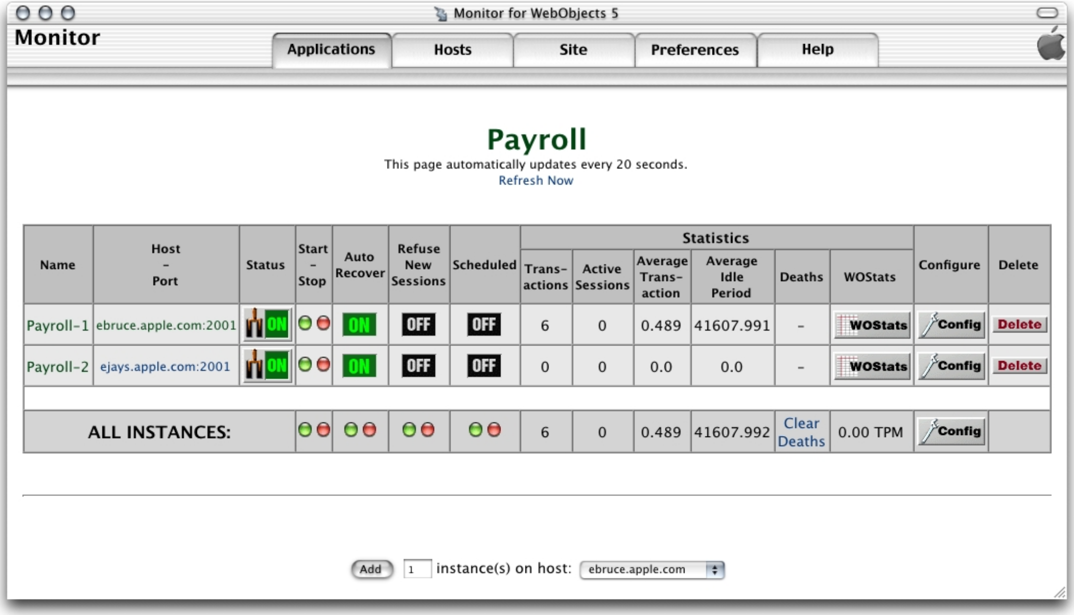
From this page, you can perform several tasks:

- **Connect to an instance of an application.** The entry page of the application is displayed in a new Web browser window.
- **Display the application detail page of an application.** For details on the information provided, see "Adding an Application" (page 76)
- **Delete an application.** You confirm that you really want to delete the application (including all of its instances) in a confirmation page before the deletion takes place.

The Application Detail Page

Figure 7-2 depicts the application detail page.

Figure 7-2 The Application Detail page



Payroll

This page automatically updates every 20 seconds.

Refresh Now

Name	Host - Port	Status	Start - Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics						Configure	Delete
							Trans-actions	Active Sessions	Average Trans-action	Average Idle Period	Deaths	WOStats		
Payroll-1	ebruce.apple.com:2001	ON	🟢🔴	ON	OFF	OFF	6	0	0.489	41607.991	-	WOStats	Config	Delete
Payroll-2	ejays.apple.com:2001	ON	🟢🔴	ON	OFF	OFF	0	0	0.0	0.0	-	WOStats	Config	Delete
ALL INSTANCES:			🟢🔴	🟢🔴	🟢🔴	🟢🔴	6	0	0.489	41607.992	Clear Deaths	0.00 TPM	Config	

Add 1 instance(s) on host: ebruce.apple.com

The application's name is displayed centered and in bold letters. When there are active instances, it is a link to the application through the HTTP adaptor; the request is load balanced. (For this to work, the HTTP adaptor URL property must be set; see "Configuring Sites" (page 92) for details. If no value has been entered for the property, the default URL (`http://localhost/cgi-bin/WebObjects/`) is used instead.) The URL used to connect to the application looks similar to the following:

```
http://localhost/cgi-bin/WebObjects/Payroll
```

The following list describes the performance-related information shown on the application detail page in the Statistics columns:

- **Transactions:** The number of requests the instance has received since it was started.
- **Active Sessions:** The number of active sessions (users) currently maintained by the instance.

- **Average Transaction:** Average time, in seconds, the instance has taken to process requests.
- **Average Idle Period:** The average time, in seconds, that the instance is idle (average time between requests).
- **Deaths:** The number of unexpected failures or deaths the instance has had since it was started. These exclude scheduled shutdowns or manual shutdowns through Monitor.
- **WOStats:** Click to display the statistics page for the instance in a new Web browser window. See "The Instance Statistics Page" (page 104) for details.

  Before you can view the instance statistics page, you have to enter the password you set on the instance configuration page. See "Setting a Password for the Instance Statistics Page" (page 90) for details.

The row with the caption ALL INSTANCES contains application-wide performance information, including the average number of transactions processed per minute (TPM).

The Instance Statistics Page

You can access the instance statistics (WOStats) page of an instance through Monitor or directly through a Web browser.

To use Monitor, go to the application detail page and click WOStats for the instance whose statistics you want to view.

To use your Web browser, access the following URL:

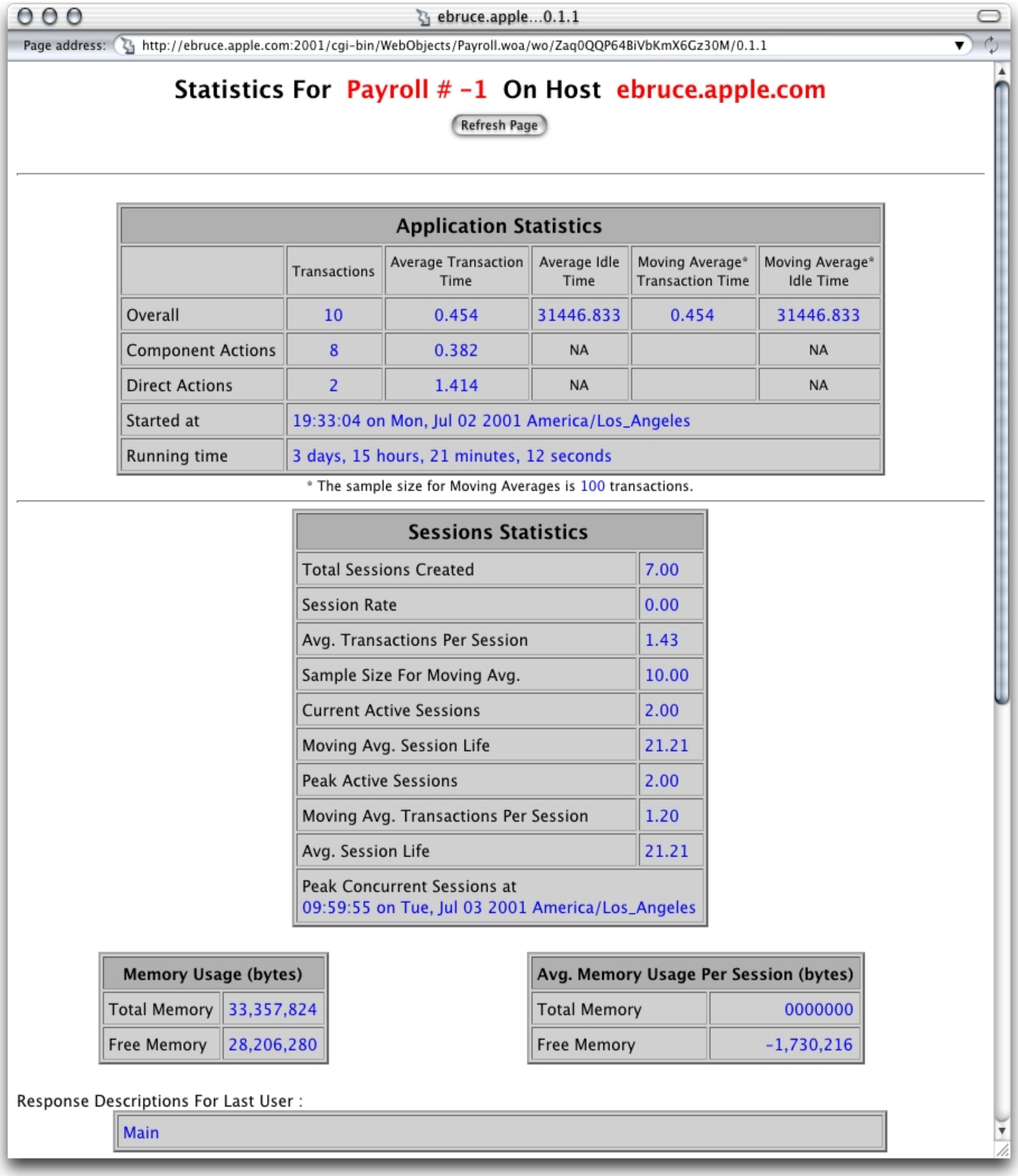
```
http://myhost/cgi-bin/WebObjects/MyApp.woa/wa/WOStats
```

If there are multiple instances, specify the instance number as well:

```
http://myhost/cgi-bin/WebObjects/MyApp.woa/1/wa/WOStats
```

Figure 7-3 and Figure 7-4 (page 106) show the information the instance statistics page provides.

Figure 7-3 The instance statistics page—part 1 of 2



http://ebruce.apple.com:2001/cgi-bin/WebObjects/Payroll.woa/wo/Zaq0QQP64BiVbKmX6Gz30M/0.1.1

Statistics For Payroll # -1 On Host ebruce.apple.com

Application Statistics					
	Transactions	Average Transaction Time	Average Idle Time	Moving Average* Transaction Time	Moving Average* Idle Time
Overall	10	0.454	31446.833	0.454	31446.833
Component Actions	8	0.382	NA		NA
Direct Actions	2	1.414	NA		NA
Started at	19:33:04 on Mon, Jul 02 2001 America/Los_Angeles				
Running time	3 days, 15 hours, 21 minutes, 12 seconds				

* The sample size for Moving Averages is 100 transactions.

Sessions Statistics	
Total Sessions Created	7.00
Session Rate	0.00
Avg. Transactions Per Session	1.43
Sample Size For Moving Avg.	10.00
Current Active Sessions	2.00
Moving Avg. Session Life	21.21
Peak Active Sessions	2.00
Moving Avg. Transactions Per Session	1.20
Avg. Session Life	21.21
Peak Concurrent Sessions at 09:59:55 on Tue, Jul 03 2001 America/Los_Angeles	

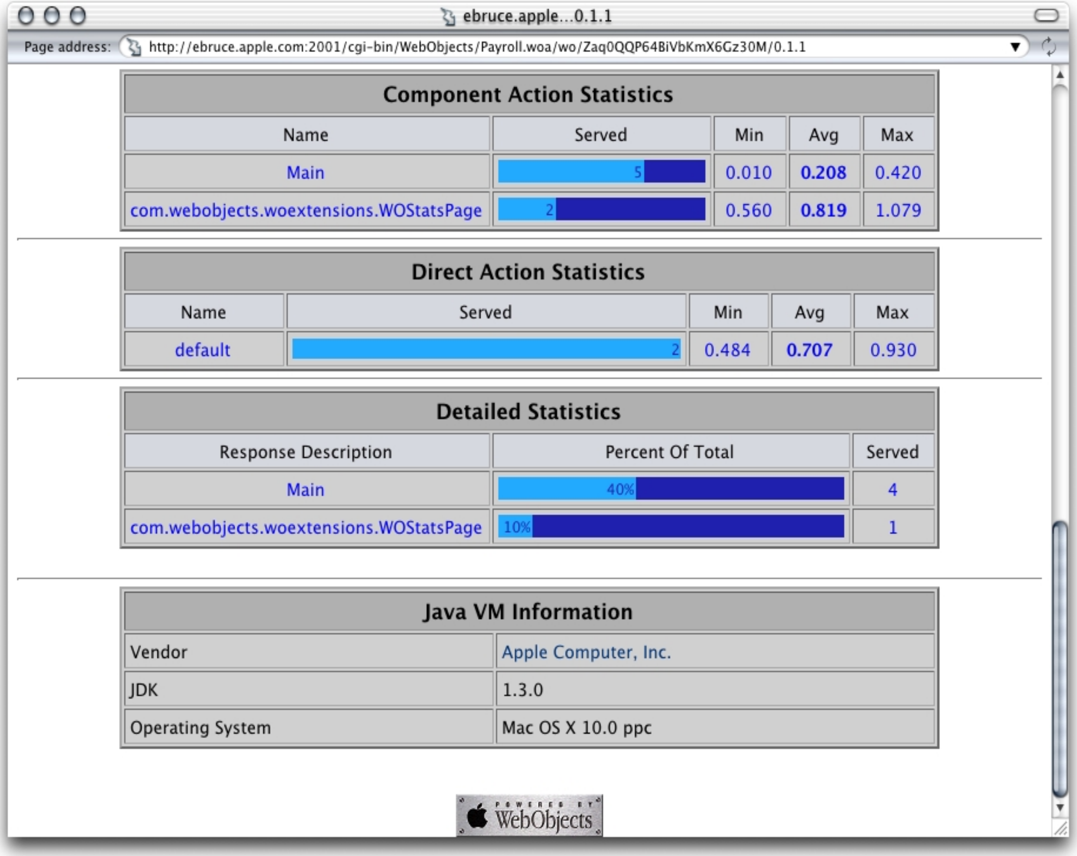
Memory Usage (bytes)	
Total Memory	33,357,824
Free Memory	28,206,280

Avg. Memory Usage Per Session (bytes)	
Total Memory	0000000
Free Memory	-1,730,216

Response Descriptions For Last User :

Main

Figure 7-4 The instance statistics page—part 2 of 2



http://ebruce.apple.com:2001/cgi-bin/WebObjects/Payroll.woa/wo/Zaq0QQP64BiVbKmX6Gz30M/0.1.1

Component Action Statistics

Name	Served	Min	Avg	Max
Main	5	0.010	0.208	0.420
com.webobjects.woextensions.WOStatsPage	2	0.560	0.819	1.079

Direct Action Statistics

Name	Served	Min	Avg	Max
default	2	0.484	0.707	0.930

Detailed Statistics

Response Description	Percent Of Total	Served
Main	40%	4
com.webobjects.woextensions.WOStatsPage	10%	1

Java VM Information

Vendor	Apple Computer, Inc.
JDK	1.3.0
Operating System	Mac OS X 10.0 ppc

Logging and Analyzing Application Activity

WebObjects applications can record information in a log file that can be analyzed by a Common Log File Format (CLFF) standard analysis tool. Applications do not maintain this log file by default; log-file recording must be enabled through an application's code. When enabled, the application records a list of components accessed during each session. By default, only component names are recorded, but you may add more information.

Logging and Analyzing Adaptor Activity

To enable adaptor logging, you create a file called logWebObjects in the temporary directory of the computer where the Web server runs. When logging is enabled, the adaptor logs its activity in a file called WebObjects.log in the temporary directory. Logging adaptor activity significantly decreases performance. Use this feature only as a troubleshooting aid; do not use it during regular deployment.

The location of the temporary directory depends on the platform:

- Mac OS X Server and Solaris: /tmp
- Windows 2000: The directory indicated by the TEMP environment variable.

Creating the Adaptor Log File

On UNIX-based platforms, do the following to create the logWebObjects file (you must have root privileges):

1. Start a command-shell window.
2. Set the working directory to the temporary directory.
3. Enter the following command:

```
touch logWebObjects
```

On Windows 2000, create a blank file using a text editor and save it as logWebObjects in the temporary directory.

You can use the tail (UNIX) or type (Windows 2000) commands to display the adaptor log file in your console.

On UNIX-based systems, use the following:

```
tail -f WebObjects.log
```

On Windows 2000, type the following in a DOS prompt:

```
type WebObjects.log
```

Analyzing the Adaptor Log File's Contents

You can analyze the information in the log to find out such things as which applications are being requested, which applications are being auto-started, and what the HTTP headers of requests are. You can also use the log to verify that the HTTP adaptor is properly configured for load balancing.

The following excerpt includes an error message that indicates that an instance of Payroll wasn't running when a request for it came in:

```
Info: <WebObjects Apache Module> new request: /cgi-bin/WebObjects/Payroll
Debug: App Name: Payroll (7)
Info: Specific instance Payroll: not found. Reloading config.
```

After Payroll is started, the same request produces the following log-file entries:

```
Info: New response: HTTP/1.0 200 Apple WebObjects
Info: ac_newInstance(): added Payroll:2 (2001)
Info: V4 URL: /cgi-bin/WebObjects/Payroll
Info: loadaverage: selected instance at index 4
Info: Selected new app instance at index 4
Debug: Composed URL to '/cgi-bin/WebObjects/Payroll.woa/1'
Info: New request is GET /cgi-bin/WebObjects/Payroll.woa/1 HTTP/1.0

Info: Sending request to instance number 1, port 2001
Info: Trying to contact Payroll:1 on (2001)
Info: attempting to connect to ebruce.apple.com on port 2001
Info: Created new pooled connection [1] to ebruce.apple.com:2001
Info: Using pooled connection to ebruce.apple.com:2001
Info: Payroll:1 on (2001) connected [pooled: Yes]
Info: Request GET /cgi-bin/WebObjects/Payroll.woa/1 HTTP/1.0
 sent, awaiting response
Debug: ac_readConfiguration(): skipped reading config
Info: New response: HTTP/1.0 200 Apple WebObjects
Info: Payroll 1 load avg = 1
Info: received ->200 Apple
```

Improving Performance

Performance is a major concern of website administrators. This section provides a list of areas to check to achieve the maximum performance possible.

- Configure your operating system so that it delivers the highest performance for your needs. Check your operating system's documentation and your Web server's documentation for performance-tuning information.
- When possible, use an API-based HTTP adaptor instead of a CGI adaptor.
- Make sure that the applications are written to perform optimally.

  The WebObjects developer documentation contains coding suggestions that help improve the performance of WebObjects applications.
- Enable component-definition caching for all applications.

  When applications are deployed, component-definition caching should be enabled so that each component's HTML and declarations files are parsed only once per session.
- Shut down and restart application instances periodically.

  Scheduling instances to shut down and restart periodically increases application performance and reliability by reducing the effects of memory leaks. For more on scheduling, see "Scheduling Settings" (page 117).

  If your applications use custom scheduling algorithms to shut down themselves, you should not use Monitor's scheduling feature. Instead, just use Monitor's auto-recover feature. For more information, see "Auto Recover" (page 114).
- Consider changing the physical configuration of your system.

  Determine the size of a single application instance (you can find this data on the application's instance statistics page) and multiply it by the number of instances you intend to run on a given computer. (For information on the instance statistics page, see "The Instance Statistics Page" (page 104).) The result is the amount of physical memory needed for that application. You have to add the memory required by the operating system, Web server, and any other applications that run constantly on the computer. The result is the amount of physical memory that should be installed on the computer.

- Try to reduce the size of the application instance by limiting the amount of state that it stores. Set the session timeout value to ensure that sessions expire after a reasonable length of time. See "WOSessionTimeOut" (page 121) for details on setting the session timeout interval for application instances.
- Make sure that all static content is served by the Web server, not your application.

If you use WebObjects Deployment mainly to deploy applications that access a datasource, you'll achieve the best performance with a dedicated datasource server and a separate server for WebObjects applications.

Deployment Settings Reference

Monitor offers an interface that you use to set the values of many configuration properties of your site. But if you prefer to use your command shell editor, there are command-line arguments available that you can use to set properties for each application instance, including Monitor and wotaskd processes.

The following sections describe the deployment properties you can set through Monitor or the command line. It contains two major sections:

- "Application Configuration Properties" (page 111) explains the properties you can set through Monitor.
- "Command-Line Arguments" (page 118) describes the function of the command-line arguments used when you start application instances from the command line (or use an application instance's Additional Arguments property).

Application Configuration Properties

There are several settings that Monitor, wotaskd, and application instances use in a deployment. You can set the value of these properties in Monitor, in the command line, or in an application's `Properties` file (which you would have to create and place in the `Contents/Resources` directory of the application). The following sections describe all the settings and their function.

Application Settings

The following sections describe the properties that apply to all the instances of an application. In Monitor, you set the value of these properties in the Application Settings section of the application configuration page.

Adaptor

The default adaptor class an instance uses. (This is not the HTTP adaptor used by the Web server.) You use this if the application defines a subclass of the WOAdaptor class to be used to create adaptor objects (instead of using the WOAdaptor class itself).

Adaptor Threads

The number of worker threads that the adaptor creates to handle requests to the application. It applies only to adaptor objects of the WODefaultAdaptor class in WebObjects 4.5.

Listen Queue Size

Determines the depth of the listen queue. While the instance handles a request, the socket buffer can hold as many additional requests as this setting indicates before it starts refusing them. If you expect spikes in the traffic level of a specific application, consider increasing the value of this property. While it does not necessarily improve performance, or allow the instance to process more requests at sustained high loads, it may reduce the number of times an application's user has to retry to send a particular request during high-traffic periods.

Keep in mind that if an application instance's listen queue size becomes full and a request is refused by it, the request does not get redirected to another instance with space left over in its queue. The client will have to resend the request.

Maximum Adaptor Threads

The maximum number of worker threads the HTTP adaptor creates to handle requests to the application. It applies only to adaptor processes of the WODefaultAdaptor class in WebObjects 5. The purpose of these threads is to process TCP or UDP packets; they have nothing to do with request processing.

Minimum Adaptor Threads

The initial number of worker threads the adaptor creates to handle requests to the application. It applies only to adaptor processes of the WODefaultAdaptor class in WebObjects 5.

Name

This property is used by the adaptor to implement load balancing. The adaptor can load-balance only between instances with the same application name. The property can be used to create groups of instances, even when the instances share the same executable file. This argument is set automatically for instances started by wotaskd.

Phased Startup

If this option is selected, when a wotaskd process starts up, the instances that it manages that have Auto Recover selected are started one-at-a-time instead of all at once. This option is selected by default. For more information, see "Auto Recover" (page 114).

Starting Port

The first port number to try to assign to a new instance of the application. If the port is in use by another instance (of any application), the next nonassigned port number is used.

Time Allowed for Startup

The number of seconds Monitor waits for an instance to start before determining that the instance failed to start.

Instance Settings

The following sections explain the properties that can be set in Monitor in the New Instance Defaults section of the application configuration page and in the Instance Settings section of the instance configuration page.

Additional Arguments

Additional information to send to the instance when it's started.

In particular, you may want to use the following additional arguments:

- `WOSessionTimeOut` (For details, see "WOSessionTimeOut" (page 121).)
- `WOStatisticsPassword` (For details, see "WOStatisticsPassword" (page 121).)

Auto-Open In Browser

Determines whether the instance automatically opens a Web browser window to the application's URL (starting up the browser if necessary).

Auto Recover

Indicates whether instances are restarted automatically whenever they become unresponsive or are manually shut down.

Caching Enabled

Determines whether the instance caches component definitions instead of parsing HTML code and declaration files each time a request is processed. (A component is a Web page or a portion of one.)

Debugging Enabled

Determines whether application instances print debugging messages to the standard error stream during startup.

ID

The instance's identification number. Must be unique for the load-balancing process to operate correctly. (This property is not available in the New Instance Defaults section.)

Lifebeat Interval

Determines the interval, in seconds, between lifebeats.

Minimum Active Sessions

The minimum number of active sessions an instance must have before it can be terminated by Monitor. (A session is an object of the WOSession class that stores user-state information.) See "Scheduling Settings" (page 117).

Output Path

Allows you to redirect the instance's standard output and standard error streams to the directory specified. The file generated is named as follows:

```
<application name>-<instance ID>
```

Path

Location of the launch script file for the application. For example, `MyApp` for Mac OS X and Solaris, and `MyApp.cmd` for Windows 2000.

Port

The port on which the instance runs. (This property is not available in the New Instance Defaults section.)

Email Notification Settings

You can tell Monitor to send a message to one of more email addresses when instances of the application terminate unexpectedly. (Manual or scheduled instance shut downs are not considered unexpected terminations.) You specify the value of these settings in the Email Notifications section of the Site page.

Load Balancing and Adaptor Settings

The following sections explain the properties related to load balancing and other adaptor settings. In Monitor, you specify the value of these settings in the HTTP Adaptor Settings section of the Site page and in the Load Balancing and Adaptor Settings section of the application configuration page.

Connect Timeout

The length of time, in seconds, before the adaptor gives up connecting to an instance.

Connection Pool Size

The maximum number of simultaneous connections the adaptor should keep open for each configured instance.

Dormant

The number of times the adaptor skips an instance of the application before trying again.

Load-Balancing Scheme

The load-balancing method used by the adaptor for instances of the application. The options provided by WebObjects are Round-Robin, Random, and Load Average. You can also use a custom load balancer by choosing the Custom option and entering the load balancer's name.

Receive Buffer Size

The size, in bytes, of the TCP socket receive buffer that's used for adaptor-to-instance communication.

Receive Timeout

The length of time, in seconds, the adaptor waits for a response from an instance of the application before giving up.

Redirection URL

The URL that the user is redirected to when an instance fails to respond to a direct request.

Retries

The number of times a request is retried (trying several instances) if a communications failure occurs before an error page is returned to the Web server.

Send Buffer Size

The size, in bytes, of the TCP socket send buffer that's used for adaptor-to-instance communication.

Send Timeout

The length of time, in seconds, the adaptor attempts to send data to an instance of the application before giving up.

URL Version

The WebObjects version to use for URL parsing and formatting. All WebObjects 4, 4.5, and 5 applications use version 4 URLs by default.

Scheduling Settings

These properties determine when Monitor restarts application instances. Restarting instances regularly helps you deploy highly reliable sites. You set the values of scheduling properties in the Scheduling section of the application configuration page.

Is Scheduled

Determines whether the schedule defined for a particular instance is active.

Graceful Scheduling

Determines whether an instance is shut down gracefully or immediately at its scheduled shut-down time. During a graceful shutdown, the instance does not create sessions for new users (they are automatically directed to other available instances, if any). Existing sessions remain active until they time out or the user logs out. When the number of active sessions drops to the value set for the Minimum

Active Sessions, the instance is restarted. When Graceful Scheduling is not selected for the instance, it is restarted immediately, terminating active sessions. See "Minimum Active Sessions" (page 115) for more information.

Types of Schedule

There are three types of schedule available for instances: hourly, daily, and weekly.

- **Hourly:** The instance is restarted after a certain number of hours from a particular hour.
- **Daily:** The instance is restarted at a particular hour every day.
- **Weekly:** The instance is restarted a particular hour on a specific day of the week.

Command-Line Arguments

The following sections describe the command-line options available to fine-tune your deployment. You specify command-line options using the following format:

```
<appName> -<optionName> <value> -<optionName> <value> ...
```

There are three types of command-line options: general, wotaskd and Monitor specific, and wotaskd specific.

General Command-Line Arguments

The sections below describe the command-line options that apply to Monitor processes, wotaskd processes, and application instances.

WOApplicationName

Value format: *string*.

Default value: Name of executable file.

Description: This setting is used by the adaptor to implement load balancing. The adaptor can load-balance only between instances with the same application name. This setting can be used to create groups of instances, even when the instances share the same executable file. This argument is set automatically for instances started by wotaskd.

WOHost

Value format: *string*.

Default value: None.

Description: Specifies the network interface that an instance binds to. This argument should only be used on hosts with multiple network interfaces (IP addresses).

WOLifebeatDestinationPort

Value format: *number*.

Default value: `1085`.

Description: Determines the port that instances send lifebeats to (should be the port that the wotaskd process that overlooks the instance runs on). The value of this property on an application instance must be the same as the value of the property for the Monitor process used to manage the instance. The wotaskd process sets this argument to the value of its `WOPort` for instances that it starts.

See: "WOPort" (page 120).

WOLifebeatEnabled

Value format: {`true`, `false`}.

Default value: `true`.

Description: Determines whether the instance sends lifebeats.

WOLifebeatInterval

Value format: *number*.

Default value: `30`.

Description: Determines the interval, in seconds, between lifebeats.

WONoPause

Value format: {`true`, `false`}.

Default value: `false`.

Description: In Windows 2000 launch scripts, if an error occurs during script execution, the process idles because the message "Press any key..." is displayed on the console and the system awaits a keypress. To avoid this behavior, set this property to `true`.

WOOutputPath

Value format: *path*.

Default value: `/dev/null`.

Description: Allows you to redirect the instance's standard output and standard error streams to the directory specified. The file generated is named as follows:

```
<application name>-<instance ID>
```

WOPort

Value format: *integer*.

Default value: `1085`.

Description: The port that the instance runs on.

See: "WOLifebeatDestinationPort" (page 119).

WORecordingPath

Value format: *path*.

Default value: None.

Description: Specifies the path for the file that stores each request and response made in a session.

WOSessionTimeOut

Value format: *number*.

Default value: `3600`.

Description: Specifies the time, in seconds, that passes after the last request is processed before the session times out.

WOStatisticsPassword

Value format: *text*.

Default value: None.

Description: Specifies the password that must be entered to gain access to the instance statistics (WOStats) page of an application instance.

Monitor and wotaskd Command-Line Arguments

There's one command-line option that applies to both Monitor and wotaskd processes but not to application instances.

WODeploymentConfigurationDirectory

Value format: *path*.

Default value: `/Library/WebObjects/Configuration`.

Description: Each wotaskd process writes its configuration to a file called `SiteConfig.xml` in the directory specified here. (The HTTP adaptor configuration file is also written to this directory.) This argument, in conjunction with `WOLifebeatDestinationPort`, allows you to run multiple wotaskd processes on a single machine.

See: "WOLifebeatDestinationPort" (page 119).

wotaskd Command-Line Arguments

These are command-line arguments that apply only to wotaskd processes.

WOAssumeApplicationIsDeadMultiplier

Value format: *integer*.

Default value: `4`.

Description: Used to determine the number of seconds that the wotaskd process waits without receiving a status message from an instance before considering it dead. It species a multiplier against `WOLifebeatInterval`. If `WOLifebeatInterval` is `30`, a wotaskd process waits 120 seconds from the last status message before determining that an instance is dead.

WOMulticastAddress

Value format: *ip-address*.

Default value: `239.128.14.2`.

Description: Sets the IP address that the wotaskd process listens to for multicast requests from the HTTP adaptor.

WORespondsToMulticastQuery

Value format: {`true`, `false`}.

Default value: `false`.

Description: Determines whether the wotaskd process responds to multicast queries from the HTTP adaptor.

WOSavesAdaptorConfiguration

Value format: {`true`, `false`}.

Default value: `false`.

Description: Determines whether the wotaskd process generates an HTTP adaptor configuration file.

A P P E N D I X A

Special Deployment Issues

The following sections explain special issues to address when deploying WebObjects applications.

Deployment Issues With Java Client Applications

A WebObjects application's developer can produce applications of two types:

- **HTML-based applications** on which the user interface elements are produced using HTML code.
- **Java Client applications**, which use Sun's Swing technology to produce a user interface that is more appealing and more efficient than HTML-based interfaces. For more information on Java Client applications see the WebObjects Java Client documentation, available at www.apple.com/developer.

There are two main issues that you should keep in mind when you deploy and administer Java Client applications:

- **session timeout**

  Java Client applications offer the user an interface that is very similar to the one offered by regular desktop applications. Therefore, they expect Java Client applications to behave in a way similar to their desktop applications.

  One of the main differences between a desktop application and a Java Client application is that Java Client applications open a connection to a server-side application. This connection expires after a certain period of inactivity. By default, the timeout period is 30 minutes. This may not be enough time for an application user that launches the application, goes to lunch, and returns to work 45 minutes later. When the user tries to use the application (which is still running on her computer), she will see a dialog that indicates that her session has timed out. In addition, any changes that were not saved are lost.

- **traffic level**

  In HTML-based applications, the packets sent between the Web browser and the Web server tend to be large. However, the size of the packets doesn't vary much (the server always sends the entire page to the browser). In Java Client applications, the packets sent by the server during application startup can be large (the entire application or part of it is downloaded); subsequent packets are relatively small (user-entered data and search results, for example).

Deploying WebObjects 4.5.1 and WebObjects 5 Applications

If you want to deploy WebObjects 4.5.1 and WebObjects 5 applications, you must use either the WebObjects 4.5.1 or the WebObjects 5 versions of Monitor and wotaskd. You cannot use the wotaskd in WebObjects 4.5.1 with the Monitor included in WebObjects 5.

Glossary

API-based adaptor HTTP adaptor based on APIs specific to a particular Web server. It allows CGI-like tasks to run as part of the main server process, avoiding the creation and termination of a process for each request. See **API-based adaptor**; **CGI adaptor**; **HTTP adaptor**.

application host A computer capable of running application instances.

bundle In Mac OS X systems, a bundle is a directory in the file system that stores executable code and the software resources related to that code. The bundle directory, in essence, groups a set of resources in a discrete package.

CGI (Common Gateway Interface) A standard for communication between external applications and information servers, such as HTTP servers.

CGI adaptor HTTP adaptor that uses the Common Gateway Interface (CGI) to translate requests from a Web server into requests to an application instance, and responses from an application instance to responses to the Web server. The Web server creates a new CGI process to handle each request. See **API-based adaptor**; **HTTP adaptor**.

component An object (of the WOComponent class) that represents a Web page or a reusable portion of one.

datasource adaptor A mechanism that connects your application to a particular database server. For each type of server you use, you need a separate adaptor. WebObjects provides an adaptor for databases conforming to JDBC. See also **JDBC adaptor**.

framework A type of bundle that packages a dynamic shared library with the resources that the library requires, including header files and reference documentation.

HTTP The client-server TCP/IP protocol used on the Web for the exchange of HTML documents.

HTTP adaptor A process (or a part of one) that connects WebObjects applications to a Web server. See also **Web server**.

Java Client A WebObjects development approach that allows you to create graphical user interface applications that run on the user's computer and communicate with a WebObjects server.

JDBC (Java Database Connectivity) An interface between Java platforms and databases.

JDBC adaptor A datasource adaptor that allows WebObjects applications to connect to JDBC-compliant database management systems. See **datasource adaptor**.

lifebeat Status message sent by WebObjects applications to wotaskd to report their activity. The four types of lifebeat messages are has started, is alive, will stop, and will crash.

load balancing Technique used to distribute user-load among the instances of an application. When multiple instances of an application are running and a new user accesses the application, the WebObjects adaptor uses one of several algorithms to determine which instance to forward the request to.

loopback Mechanism that allows you to open a connection to a machine that does not go over the network.

Monitor A tool used to configure and maintain deployed WebObjects applications capable of handling multiple applications, application instances, and applications hosts at the same time.

Project Builder A tool used to manage the development of a WebObjects application or framework.

session A period during which access to a WebObjects application and its resources is granted to a particular client (typically a browser). Also an object (of the WOSession class) representing a session.

SMTP (Simple Mail Transfer Protocol) A protocol used to transfer email between computers, usually over Ethernet.

socket Mechanism for creating a virtual connection between processes. It interfaces standard I/O with network communication facilities. A socket address consists of a port number and an IP address.

UDP (User Datagram Protocol) Lightweight and efficient connectionless datagram transport protocol. Used to send self-routing data throughout a network.

Web server An application that serves Web pages to Web browsers using the HTTP protocol. In WebObjects, the Web server lies between the browser and a WebObjects application. When the Web server receives a request from a browser, it passes the request to the WebObjects adaptor, which generates a response and returns it to the Web server. The Web server then sends the response to the browser. See also **HTTP adaptor**.

WebObjects Deployment Software package that allows you to deploy WebObjects applications on an intranet or the Web. You need to install a WebObjects deployment license on computers on which you want to install this package. See also **WebObjects Development**.

WebObjects Development Software package that allows you to develop WebObjects applications. It includes tools to design applications using an object-oriented approach. You need to install a WebObjects development license on computers on which you want to develop applications. See also **WebObjects Deployment**.

WOServices WebObjects service that monitors wotaskd processes. Its main duty is to monitor wotaskd and restart it if it dies or when the host is restarted. The implementation of this service is platform-dependent.

wotaskd WebObjects Deployment tool that manages the instances on an application host. It's used by Monitor to propagate site configuration changes throughout the site's application hosts. See also **Monitor**.

Glossary