

**SUPPORTING
WEB SERVERS**

TK 5105

.888

.D36

2001

Copy 1



S U P P O R T I N G

Web Servers

Interactive Workbook

- ◆ The first hands-on networking workbook for Web professionals
- ◆ Maximize the performance and reliability of your site's Internet connection
- ◆ Internet applications, protocols, security, scripting, and more
- ◆ Tips on choosing and implementing next-generation technologies



BENAY DARA-ABRAMS • DREW DARA-ABRAMS
TREVOR PEACE • BEBO WHITE

Supporting Web Servers



THE ADVANCED WEBSITE ARCHITECTURE SERIES

DESIGNING WEB INTERFACES

Michael Rees
Andrew White
Bebo White

SUPPORTING WEB SERVERS

Benay Dara-Abrams
Drew Dara-Abrams
Trevor Peace
Bebo White

ANALYZING E-COMMERCE & INTERNET LAW

J. Dianne Brinson
Benay Dara-Abrams
Drew Dara-Abrams
Jennifer Masek
Ruth McDunn
Bebo White

Supporting Web Servers

Benay Dara-Abrams

Drew Dara-Abrams

Trevor Peace

Bebo White



Prentice Hall PTR
Upper Saddle River, NJ 07458
www.phptr.com

Library of Congress Cataloging-in-Publication Data

Dara-Abrams, Benay.

Supporting web servers / Benay Dara-Abrams, Drew Dara-Abrams
Trevor D. Peace, Bebo White.

p. cm.— (Advanced web site architecture)

ISBN 0-13-085899-4

1. Web site development. 2. Web servers. I. Dara-Abrams, Drew. II. Peace,
Trevor D. III. White, Bebo, 1945- IV. Title. V. Series.

TK5105.888 .D36 2001

005.7' 1376—dc21

00-054714

Editorial/production supervision: *Jessica Balch (Pine Tree Composition)*

Project coordination: *Anne Trowbridge*

Acquisitions editor: *Karen McLean*

Developmental editor: *Ralph Moore*

Editorial assistant: *Richard Winkler*

Manufacturing manager: *Alexis R. Heydt*

Marketing manager: *Kate Hargett*

Cover design director: *Jerry Votta*

Interior designer: *Meryl Poweski*



©2001 Prentice Hall PTR

Prentice-Hall, Inc.

Upper Saddle River, NJ 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact: Corporate Sales Department, Phone: 800-382-3419; Fax: 201-236-7141; E-mail: corpsales@prenhall.com; or write: Prentice Hall PTR, Corp. Sales Dept., One Lake Street, Upper Saddle River, NJ 07458.

All products or services mentioned in this book are the trademarks or service marks of their respective companies or organizations.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-085899-4

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Pearson Education Asia Pte. Ltd.

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

TK5105
.888
.D36
2001
Copy 1

Dedicated to my family, who believed me when I said that the Internet and the Web would change how we communicate, how we do business, and how we teach and learn.

—B.D-A.

Dedicated to my family, who help enable my dreams and never tire of listening to my grand ideas.

—D.D-A.

To Sandy.

—T.P.

My contributions to this book series are dedicated to my loving and supportive family—Nancy, Andrew, and Christopher—whose tolerance for patience is always severely tested whenever I take on a book project. We did it again!

—B.W.



CONTENTS

From the Editor	xv
Executive Foreword	xvii
Introduction	xix
About the Authors	xxii
CHAPTER 1 Networking Fundamentals for Webmasters	1
Lab 1.1 Networks and Network Characteristics	3
1.1.1 Identify Different Types of Networks	5
1.1.2 Identify Network Resource-Sharing Facilities	5
Self-Review Questions	7
Lab 1.2 The Open Systems Interconnection (OSI) Reference Model	8
1.2.1 Identify Where HTTP Fits in the OSI Model	11
1.2.2 Give Examples of Protocols on Different Layers	11
Self-Review Questions	12
Lab 1.3 Internetworks and Network Topologies	14
1.3.1 Identify Various Network Topologies	16
1.3.2 Understand the Function of Internetworks	17
Self-Review Questions	19
Lab 1.4 Networks Characterized by Spatial Distance	20
1.4.1 Understand the Benefits of LAN Technology	22
1.4.2 Relate Spatial Characterization to the Internet, Intranets, and Extranets	22
Self-Review Questions	24
Lab 1.5 Intranets and Extranets	26
1.5.1 Describe the Major Components of an Intranet and an Extranet	29
1.5.2 Understand the Differences Between Intranets, Extranets, and the Internet	30
Self-Review Questions	31
Test Your Thinking	33
	vii

CHAPTER 2 Architectures, Devices, and Connections	35
Lab 2.1 Packet-Switched Networks	36
2.1.1 Describe How a Packet-Switched Network Works	37
2.1.2 Compare Packet Switching and Circuit Switching	37
Self-Review Questions	39
Lab 2.2 The Web Client/Server Model	41
2.2.1 Understand the Difference Between Client/Server, Peer-to-Peer, and Master/Slave Architectures	43
2.2.2 Appreciate the Advantage of Client/Server Architecture for the Web	43
Self-Review Questions	45
Lab 2.3 Ethernet and FDDI Networks	47
2.3.1 Understand the Advantages of Using Ethernet Technology	48
2.3.2 Compare Ethernet and FDDI Technologies	49
Self-Review Questions	50
Lab 2.4 Network Devices	52
2.4.1 Distinguish Between Routers, Bridges, and Switches	53
2.4.2 Determine Appropriate Interconnecting Devices for Your Network Architecture	54
2.4.3 Relate LAN Topologies to Switches	54
Self-Review Questions	55
Lab 2.5 Network Connections	57
2.5.1 Understand Different Modes of Access to Your Internet Service Provider (ISP)	59
2.5.2 Determine Why PPP Is Preferred over SLIP	59
Self-Review Questions	61
Test Your Thinking	62
CHAPTER 3 Internet, Network, and Transport Protocols	63
Lab 3.1 Internet and Internet Protocol (IP)	65
3.1.1 Determine the Organizations Responsible for Domain Name Registration	67
3.1.2 Register a New Domain Name	67
Self-Review Questions	68

	Contents	ix
Lab 3.2	IP Addresses	70
3.2.1	Understand the IP Addressing Scheme	73
3.2.2	Know How to Identify a Domain Name by Its IP Address	73
	Self-Review Questions	75
Lab 3.3	Domain Name System (DNS)	77
3.3.1	Understand How Domains Are Associated with IP Addresses	78
3.3.2	Know How a Domain Name Is Resolved	78
	Self-Review Questions	79
Lab 3.4	Transmission Control Protocol (TCP)	81
3.4.1	Determine the Socket Number for “Well Known Services”	84
3.4.2	Understand the Use of Ports for Services	85
	Self-Review Questions	86
Lab 3.5	MAC Addresses	87
3.5.1	Find MAC Addresses of Computers within Your LAN	90
3.5.2	Read and Interpret MAC Addresses	90
	Self-Review Questions	91
Lab 3.6	User Datagram Protocol (UDP)	93
3.6.1	Understand the Difference Between TCP and UDP Protocols	94
3.6.2	List Applications that May Use the UDP Protocol	
	and Understand Why	94
	Self-Review Questions	95
	Test Your Thinking	97
CHAPTER 4	Applications and Protocols	99
Lab 4.1	Hypertext Transfer Protocol (HTTP)	100
4.1.1	Diagram Web Page Retrieval from the Internet	104
4.1.2	Diagram Web Page Retrieval from Cache	104
	Self-Review Questions	106
Lab 4.2	SMTP, MIME, and FTP	108
4.2.1	Understand How Mail Protocols Work	109
4.2.2	Know How to Use File Transfer Protocol	110
	Self-Review Questions	111
Lab 4.3	Push and Pull Technologies	113
4.3.1	Differentiate Between Push and Pull Technologies	114

x *Contents*

4.3.2	Consider When Push Technology Is Appropriate	114
	Self-Review Questions	115
Lab 4.4	Streaming Multimedia	117
4.4.1	Know the Delivery Mechanisms Involved in Streaming Media	119
4.4.2	Determine When to Provide Streaming Multimedia Facilities	120
	Self-Review Questions	122
	Test Your Thinking	123
 CHAPTER 5 Connectivity		 125
Lab 5.1	ISDN, DSL, and T-Carrier Technologies	126
5.1.1	Differentiate Between Various Technologies Offered by Telephone Companies	129
5.1.2	Determine Availability of Network Services in Your Area	129
	Self-Review Questions	131
Lab 5.2	Wireless Networks	132
5.2.1	Differentiate Between Wireless and Wired Networks	133
5.2.2	Understand How Wireless Networks Transmit Data	133
	Self-Review Questions	136
Lab 5.3	Cable Modems	138
5.3.1	Compare Various Network Technologies and Services	139
5.3.2	Differentiate Between Technologies Offered by the Telephone Companies and by the Cable Companies	139
	Self-Review Questions	142
	Test Your Thinking	143
 CHAPTER 6 Network Security		 145
Lab 6.1	Encryption Systems	146
6.1.1	Understand the Use of Public Key Cryptography	152
6.1.2	Consider the Trade-offs in Encrypting Messages	152
	Self-Review Questions	153
Lab 6.2	Secure Internet Protocols	155
6.2.1	Understand the Function of the SSL Protocol	159
6.2.2	Develop a Privacy Statement	159
	Self-Review Questions	161

	<i>Contents</i>	xi
Lab 6.3	Firewalls and Sniffers	162
6.3.1	Understand the Functionality of a Firewall	163
6.3.2	Distinguish Between Use of Sniffers by Network Administrators and Hackers	164
	Self-Review Questions	165
Lab 6.4	Proxy Servers, Virtual Private Networks, and Smartcards	167
6.4.1	Understand the Functionality of a Proxy Server	169
6.4.2	Assess the Feasibility of a Smartcard Solution	170
	Self-Review Questions	172
	Test Your Thinking	174
CHAPTER 7	Web Server Support	175
Lab 7.1	Web Server Evaluation Issues	177
7.1.1	Understand the Basic Concepts of Web Hosting	183
	Self-Review Questions	185
Lab 7.2	Web Site Service Models	187
7.2.1	Understand the Basic Web Site Architectures	190
	Self-Review Questions	192
Lab 7.3	Supporting Multiple Servers	193
7.3.1	Choose an Appropriate Web Server Configuration	196
	Self-Review Questions	198
Lab 7.4	Server Security Issues	200
7.4.1	Understand Basic Security Risks	208
7.4.2	Implement Appropriate Security Measures	209
	Self-Review Questions	212
Lab 7.5	Document Root Taxonomies	214
7.5.1	Design Appropriate Document Root Taxonomies	216
	Self-Review Questions	218
Lab 7.6	Access Authorization, Security, and Privacy	221
7.6.1	Understand Basic Security Configuration	225
	Self-Review Questions	227
Lab 7.7	Searching and Indexing Issues	228
7.7.1	Understand Searching and Indexing Issues	234
	Self-Review Questions	236
	Test Your Thinking	237

CHAPTER 8 The Web Programming and Scripting Environment

Lab 8.1	Reasons Why Developers Write Server-Side Script	240
8.1.1	Discuss the Problems of Writing Client-Side Code	241
8.1.2	Explain the “Lowest Common Denominator” Solution	242
	Self-Review Questions	244
Lab 8.2	3-Tiered Web Application Design	245
8.2.1	Understand and Explain the Presentation Layer	246
8.2.2	Understand and Explain the Business Logic Layer	247
8.2.3	Understand and Explain the Persistence Layer	248
	Self-Review Questions	250
	Test Your Thinking	252

CHAPTER 9 Programming and Scripting in a Client/Server System

Lab 9.1	Server-side Programming	254
9.1.1	Identify Key Technologies for Processing Server-Side Code	256
9.1.2	List Advantages and Disadvantages of Server-Side Code	257
	Self-Review Questions	259
Lab 9.2	Client-Side Programming	261
9.2.1	Write Simple Client-Side JavaScript	263
9.2.2	Perform Basic Client-Side Form Validation	264
	Self-Review Questions	267
Lab 9.3	Combining Client-Side and Server-Side Scripting	269
9.3.1	Combine Server-Side and Client-Side Logic	270
9.3.2	Explain How Cookies Are Both Client- and Server-Side Objects	271
	Self-Review Questions	273
	Test Your Thinking	275

CHAPTER 10 Programming and Scripting Security Issues

Lab 10.1	Security Tradeoffs	278
10.1.1	Identify Potential Security Threats	279

	Contents	xiii
10.1.2	Find a Balance Between Restrictive and Functional Self-Review Questions	279 282
Lab 10.2	The Importance of Validating User Input	283
10.2.1	Examine User Input	284
10.2.2	Prevent Users from Entering HTML	284
10.2.3	Filter Out Malicious SQL Code Self-Review Questions	285 288
Lab 10.3	User versus Webmaster Scripting	290
10.3.1	Identify Web Applications That Are Both Fun and Safe	291
10.3.2	Identify Web Applications That Are Useful but Dangerous Self-Review Questions	291 293
	Test Your Thinking	295
CHAPTER 11	Scripting Language Evolution	297
Lab 11.1	ASP	298
11.1.1	Output Basic Variables	299
11.1.2	Code a Basic If-Then Statement	300
11.1.3	Code a While Loop Self-Review Questions	301 304
Lab 11.2	Cold Fusion	305
11.2.1	Output Variables	306
11.2.2	Code a Basic If-Then Statement	307
11.2.3	Code a Loop Self-Review Questions	308 312
Lab 11.3	PHP	313
11.3.1	Output Basic Variables	314
11.3.2	Code a Basic If-Then Statement	316
11.3.3	Code a While Loop Self-Review Questions	317 321
Lab 11.4	Language Evolution	323
11.4.1	Describe Conceptual Building Blocks	324
11.4.2	Identify Emerging and Evolving Technologies Self-Review Questions	325 327
	Test Your Thinking	329

CHAPTER 12	Supporting Supplementary Technologies	331
Lab 12.1	Internet and WWW Standardization Activities	333
12.1.1	Understand the Role of Standards and Standardization	341
12.1.2	Know Which Organizations Impact the Development of WWW Standards	342
12.1.3	Understand the Standardization Process Used by These Organizations	342
12.1.4	Understand How to Use These Organizations as an Information Resource	342
	Self-Review Questions	344
Lab 12.2	Strategies for Evaluating New Technologies	346
12.2.1	Understand the Importance of Standardization When Evaluating New Technology	352
12.2.2	Understand When New Technologies Should Be Introduced in the Web Site Life Cycle	352
	Self-Review Questions	353
Lab 12.3	New Technologies to Watch	355
12.3.1	Understand the W3C Functional Domains	371
12.3.2	Understand the Collaboration between the W3C and the IETF	371
	Self-Review Questions	372
	Test Your Thinking	374
Appendix	Answers to Self-Review Questions	375
Index		382

FROM THE EDITOR

As the Internet rapidly becomes the primary commerce and communications medium for virtually every company and organization operating today, a growing need exists for trained individuals to manage this medium. Aptly named *Webmasters*, these individuals will play leading roles in driving their organizations into the next millennium.

Working with the World Organization of Webmasters (WOW), Pearson PTR has developed two book series that are designed to train Webmasters to meet this challenge. These are *The Foundations of Website Architecture Series*, and *The Advanced Website Architecture Series*.

The Webmaster who masters the materials in these books will have working knowledge of Web site management, support, maintenance, organizational strategy, electronic commerce strategy and tools, as well as legal issues surrounding the Web. The Webmaster will be able to implement sound Web site design, navigation and HCI (Human-Computer Interaction) practices. Webmasters will also have a solid understanding of networking, Web servers, Web programming, and scripting, as well as supporting supplementary technologies.

The goal of *The Advanced Website Architecture Series* is to provide an advanced Webmaster training curriculum. *The Advanced Website Architecture Series* offers in-depth coverage of the content, business, and technical issues that challenge Webmasters.

Books in this series are:

Designing Web Interfaces
Supporting Web Servers
Analyzing E-Commerce & Internet Law

The Foundations of Website Architecture Series is designed to introduce and explain the technical, business, and content management skills that are necessary to effectively train the new Webmaster.

Books in *The Foundations of Website Architecture Series* include:

Understanding Web Development

Mastering Internet Protocols

Administrating Web Servers, Security, & Maintenance

Exploring Web Marketing & Project Management

Creating Web Graphics, Audio, & Video

Thank you for your interest in *The Advanced Website Architecture Series*, and good luck in your career as a Webmaster!

Karen McLean
Senior Managing Editor
Pearson PTR Interactive

EXECUTIVE FOREWORD

Within the next few years, you will think about the Internet in the same way you think about electricity today. Just as you don't ask a friend to "use electricity to turn on a light," you will assume the omnipresence of the Web and the capabilities that it delivers. The Web is transforming the way we live, work, and play, just as electricity changed everything for previous generations.

Every indication suggests that the explosive growth of the Web will continue. The question we need to address is, "How can we deliver the most value with this ubiquitous resource?" Today, most of the world's Web sites were created and are maintained by self-taught Webmasters. Why? Because there were limited opportunities to receive formal standards-based education. Quality, accessible, affordable education will help provide the broad range of knowledge, skills, and abilities to meet the demands of the marketplace.

Over the last three years, the World Organization of Webmasters (WOW) has worked with colleges and universities, business and industry, and its own membership of aspiring and practicing Web professionals to develop the Certified Web Professional (CWP) program. Our three-part goal is to provide:

- Educational institutions with guidelines around which to develop curricula.
- An organized way to master technical skills, content development, business proficiency and personal workplace ability.
- An assessment standard for employers to measure candidates.

The Foundations of Website Architecture Series and *The Advanced Website Architecture Series* grew organically from the communities they will serve. Written by working professionals and academics currently teaching the material, and reviewed by leading faculty at major colleges and universities and the WOW Review Board of industry professionals, these books are designed to meet the increasingly urgent need for Web professionals with expertise in three areas: technical development, design and content development, and business.

There is a huge increase in demand for qualified Web professionals, with the number of Web sites projected to grow from about 5 million today to about 25 million by the year 2002. Web professionals with business, technical, design, and

xviii *Executive Foreword*

project management skills are, and will continue to be, the most in-demand and will receive the highest compensation.

On behalf of WOW and its members, we wish you the best of success and welcome you to this exciting field.

*William B. Cullifer
Executive Director-Founder
World Organization of Webmasters
bill@joinwow.org*

INTRODUCTION

WHAT YOU WILL NEED

A networked PC with access to the Internet. The faster the connection, the less time you spend on the “World Wide Wait.”

A Web browser with as many plug-ins as you can support (to experience as much marketing media as possible) and an e-mail account. In your browser preferences, please enable cookies.

HOW THIS BOOK IS ORGANIZED

In this book, and the others in this series, you are presented with a series of interactive labs. Each lab begins with Learning Objectives that define what exercises (or tasks) are covered in that lab. This is followed by an overview of the concepts that will be further explored through the exercises, which are the heart of each lab.

Each exercise consists of either a series of steps that you will follow to perform a specific task or a presentation of a particular scenario. Questions that are designed to help you discover the important things on your own are then asked of you. The answers to these questions are given at the end of the exercises, along with more in-depth discussion of the concepts explored.

At the end of each lab is a series of multiple-choice Self-Review Questions, which are designed to bolster your learning experience by providing opportunities to check your absorption of important material. The answers to these questions appear in the Appendix. There are also additional Self-Review Questions at this book's companion Web site, found at <http://www.phptr.com/phptrinteractive/>.

Finally, at the end of each chapter you will find a Test Your Thinking section, which consists of a series of projects designed to solidify all of the skills you have learned in the chapter. If you have successfully completed all of the labs in the chapter, you should be able to tackle these projects with few problems. There are not always “answers” to these projects, but where appropriate, you will find guidance and/or solutions at the companion Web site.

The final element of this book actually doesn't appear in the book at all. It is the companion Web site, and it is located at <http://www.phptr.com/phptrinteractive/>.

This companion Web site is closely integrated with the content of this book, and we encourage you to visit often. It is designed to provide a unique interactive

online experience that will enhance your education. As mentioned, you will find guidance and solutions that will help you complete the projects found in the Test Your Thinking section of each chapter.

You will also find additional Self-Review Questions for each chapter, which are meant to give you more opportunities to become familiar with terminology and concepts presented in the publications. In the Author's Corner, you will find additional information that we think will interest you, including updates to the information presented in these publications, and discussion about the constantly changing technology Webmasters must stay involved in.

Finally, you will find a Message Board, which you can think of as a virtual study lounge. Here, you can interact with other *Advanced Website Architecture Series* readers, and share and discuss your projects.

NOTES TO THE STUDENT

This publication and the others in *The Advanced Website Architecture Series* are endorsed by the World Organization of Webmasters. The series is a training curriculum designed to provide aspiring Webmasters with the skills they need to perform in the marketplace. The skill sets included in *The Advanced Website Architecture Series* were initially collected and defined by this international trade association to create a set of core competencies for students, professionals, trainers, and employers to utilize.

NOTES TO THE INSTRUCTOR

Chances are that you are a pioneer in the education field whether you want to be one or not. Due to the explosive nature of the Internet's growth, very few Webmaster training programs are currently in existence. But while you read this, many colleges, community colleges, technical institutes, and corporate and commercial training environments are introducing this material into curriculums worldwide.

Chances are, however, that you are teaching new material in a new program. But don't fret, this publication and series are designed as a comprehensive introductory curriculum in this field. Students successfully completing this program of study will be fully prepared to assume the responsibilities of a Webmaster in the field or to engage in further training and certification in the Internet communications field.

Each chapter in this book is broken down into labs. All questions and projects have the answers and discussions associated with them. The labs and question/answer formats used in this book provide excellent opportunities for group discussions and dialogue between students and instructors. Many answers and their

discussions are abbreviated in this publication for space reasons. Any comments, ideas, or suggestions to this text and series will be would be greatly appreciated.

ACKNOWLEDGMENTS

From Benay: My sincere appreciation to my supportive family, particularly my wonderful daughter Cassie, for encouraging me to develop, teach, and write about Web technology.

From Drew: My thanks to my family, who encouraged me to take on this project.

From Bebo: I am grateful to

- Robert Cailliau and Tim Berners-Lee who let me share their visions a decade ago and have continued to support me.
- My book co-authors who worked tirelessly to make this book series the best that it could be.
- Bill Cullifer of WOW for finding in Prentice Hall, a publisher not interested in just another book series on the Web, but a series that dares to address many of the Web management issues that have been previously overlooked or ignored.
- Karen McLean of Prentice Hall for a patient, yet firm, hand that pushed this project to completion.

CHAPTER 8

THE WEB PROGRAMMING AND SCRIPTING ENVIRONMENT



Polished brass will pass upon more people than rough gold.

—Chesterfield

CHAPTER OBJECTIVES

In this chapter, you will learn about:

- ✓ Reasons Why Developers Write Server-Side Script Page 240
- ✓ 3-Tiered Web Application Design Page 245

One of the most amazing things about the Internet (and about the World Wide Web in particular) is the platform agnosticism between client and server. The details of the underlying operating system, computer hardware, or network configuration don't matter one bit, as long as both client and server conform to a standard set of protocols. For that reason, there has been a recent explosion of options for Web servers, browsers, programming languages, and scripting technologies. In this chapter, we classify Web technologies into two functional categories (server-side and client-side) and then again into three conceptual layers (presentation layer, business logic, and persistence layer). By learning to identify the right tool for the right job, you will be able to select the technologies that will best suit your deployment requirements, your development environment, and your projected scalability needs.

LAB 8.1

REASONS WHY DEVELOPERS WRITE SERVER-SIDE SCRIPT

LAB OBJECTIVES

After this lab, you will be able to:

- ✓ Discuss the Problems of Writing Client-Side Code
- ✓ Explain the "Lowest Common Denominator" Solution

Back in the "good old days" of software development (before the Web existed, but not so far back as punch cards), there were a few programming languages to choose from, and there were a few compilers for each language. Once you had selected a language and a compiler, all that was left was to write code that compiled. You had complete control over your development environment.

Writing code for the Web has always been a little different. The development pace is rapid, the number of Web standards is literally exploding, and nobody has the time to keep on top of all the various standards being debated, implemented, and deprecated. Even worse, browser support for these standards often varies widely. Each Web browser implements the various Web standards in a slightly different way. Client-side code that works fine in Internet Explorer and Opera might crash Netscape and be completely useless in Lynx (or any other text-based browser).

The problem with writing client-side code quickly became apparent: The developer has virtually no control over the execution environment. True, good developers can make educated guesses about which browsers are most likely to access their pages, but there is always some uncertainty. Trying to write code that works in all major browsers is like trying to write code that will compile in several different compilers at once (if readers are not familiar with the joys of trying to

compile code on multiple platforms, they may consider themselves extremely lucky).

Faced with a variety of incompatible browser “enhancements” and incomplete support for Web standards, one of the more prevalent (albeit disheartening) techniques of development is to write code that only uses the subset of HTML, CSS, JavaScript, and DHTML functionality that is consistently supported by the major browsers (the “lowest common denominator”). Just as prime-time television is dumbed down to appeal to the widest possible audience, compatibility on the World Wide Web often means hobbling the code to avoid breaking the oldest, least advanced browser still in wide usage. This “compromise” of practicality over functionality left many Web developers dissatisfied at least in part because it means neglecting new and exciting features that older browsers are not designed to handle.

Responding to this developer discontent, Sun made a lot of noise promoting Java, which was supposed to provide a “write once, run anywhere” approach to cross-platform coding. In practice—so the joke goes—it often worked out to be more of a “write once, debug everywhere” approach for the developer, because Java virtual machine implementations also varied somewhat from platform to platform (and from browser to browser).

About the same time that Java was appearing on the scene, CGI scripts were becoming increasingly advanced (they had been around years before Java, of course). These scripts were pieces of code that ran on the Web server, not on the user’s browser. By running code on a server, in a controlled environment, the whole question of cross-browser compatibility became moot. As long as the Web server returns nothing but HTML to the client, the Web application developer doesn’t have to worry about cross-browser quirks at all. One can almost picture Web developers breathing a sigh of relief as they began writing code in a server-side environment they could control, after struggling to produce client-side code that had to work in a variety of browser environments out of their control.

LAB 8.1 EXERCISES

8.1.1 DISCUSS THE PROBLEMS OF WRITING CLIENT-SIDE CODE

- a) Which major browser do you think supports more than 75% of all client-side JavaScript features?

242 *Lab 8.1: Reasons Why Developers Write Server-Side Script*

- b) When can you depend on client-side scripting for mission-critical logic?

- c) Suppose there is a form on the page, and that form needs to be checked for mistakes (letters in number boxes, blank fields, invalid dates, etc.) before the information can be entered in the database. Do you want to write client-side or server-side code to do that error checking? Why?

8.1.2 EXPLAIN THE “LOWEST COMMON DENOMINATOR” SOLUTION

Imagine that you’re working on a Web site for a client. He wants to make sure that anyone with Internet access can get full functionality out of his Web site. In fact, he’ll be viewing your work via a dialup connection and AOL.

- a) What design limitations are you under?

- b) How much functionality do you think you can build in?

LAB 8.1 EXERCISE ANSWERS**LAB
8.1****8.1.1 ANSWERS**

- a) Which major browser do you think supports more than 75% of all client-side JavaScript features?

Answer: Sorry. This was a trick question. No one has written a browser that supports anywhere near 75% of all the features and functionality defined in JavaScript. Stay on the beaten path and you'll be okay, but if you go wandering off into some of the darker corners of JavaScript, you may rapidly find yourself running into subtle bugs caused by quirky or incomplete browser support.

- b) When can you depend on client-side scripting for mission-critical logic?

Answer: Never. If there is a feature of your publicly accessible Web site that is absolutely critical—that your site cannot work without—under no circumstances should you rely on client-side routines to perform that logic.

As a side note to that answer, there are times, especially when developing Web applications for a company's internal use, when the developer is in a position to standardize the browser configuration for an application. If you know that all of the visitors to your page will be using Browser X with JavaScript enabled, then you can feel confident in making the most of the client-side code supported by Browser X. When writing Web pages for general Internet use, however, you can make no such assumptions.

- c) Suppose there are forms on the page, and those forms need to be checked for mistakes (letters in number boxes, blank fields, invalid dates, etc.) before the information can be entered in the database. Do you want to write client-side or server-side code to do that error checking? Why?

Answer: You want to do server side checking at the bare minimum. There are times when it's nice to have a client-side box pop up informing the user of an error before the form is submitted, but don't rely exclusively on JavaScript error checking. The reason? Users are allowed to turn off JavaScript from within their browsers, and they often do.

Whenever you insert any information into a database, it's a good idea to error-check the data before adding it. We will touch on this topic again in Chapter 10 on security issues, but it bears repeating. Speaking from experience, it is incredibly embarrassing to have your database-driven application crash due to relational inconsistencies after someone turned off JavaScript in his or her browser and submitted an incomplete form that bypassed your intricately written JavaScript error-checking routines, inserting bad data into your database.

8.1.2 ANSWERS

- a) What design limitations are you under?

Answer: You are under some serious design restrictions. Avoid any unnecessary client-side code and try to stick to straight HTML. Also remember to keep the graphics light: Web developers (designers especially) often forget the pain of a dial-up connection.

- b) How much functionality do you think you can build in?

Answer: A surprising amount, provided you build most, if not all of all the application's functionality into server-side code. Rumor has it that amazon.com is completely functional through a 2.0 browser. That means that Amazon uses nothing more advanced than HTML tables on the client side. It's completely possible to build a functional dynamic Web site that is compatible with all 3.0 browsers. You will just need to avoid adding functionality through client-side scripting.

It used to be that writing for the lowest common denominator meant that everything on the site had to work properly in a 3.0 browser. More recently, however, it is not considered unreasonable to build sites for browsers comparable to (or better than) Internet Explorer or Netscape 4.0.

LAB 8.1 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

- 1) As long as a Web server returns nothing but HTML to a client, the Web application developer doesn't have to worry about cross-browser quirks at all.
 - a) ☐ True
 - b) ☐ False
- 2) Both Netscape Navigator and Internet Explorer, versions 5.0 and higher, support more than 75% of all client-side JavaScript features.
 - a) ☐ True
 - b) ☐ False
- 3) Even the very best developers can make educated guesses about which browsers are most likely to access their pages, but there is always some uncertainty.
 - a) ☐ True
 - b) ☐ False

Quiz answers appear in the Appendix, Section 8.2.

LAB 8.2

3-TIERED WEB APPLICATION DESIGN

**LAB
8.2**

LAB OBJECTIVES

After this lab, you will be able to:

- ✓ Understand and Explain the Presentation Layer
- ✓ Understand and Explain the Business Logic Layer
- ✓ Understand and Explain the Persistence Layer

When you start coding Web applications more advanced than pages that know when to say “good morning” and when to say “good afternoon,” you will rapidly discover that scripting languages are relatively simple and extremely flexible. So flexible, in fact, that there aren’t any clear guidelines for how code should be written. Sure, the syntax is straightforward, but actually organizing a single page can get to be a chore. Getting five pages to talk to each other can be a jumbled mess. And getting twenty or thirty different pages to all work together to form one cohesive Web application can be a nightmare . . . unless you impose some artificial structure onto the process.

Most Web design is broken down into three layers, or tiers. You may have heard it referred to as 3-tiered Web design or possibly even N-tiered application design (“N-tiered” means that the application can be broken down into an arbitrarily large number of conceptual pieces, with the possibility that each piece could reside on its own machine). The three tiers are as follows: the presentation layer, the business logic layer, and the persistence layer.

The presentation layer is the front end of the site. The look and feel of the site is within the domain of presentation. The user interface also falls into the presenta-

tion layer, as does personalization code. In fact, pretty much anything the user interacts with, clicks on, or reads from is within the presentation layer.

The technologies underlying the presentation layer are Web scripting languages. These languages are all interpreted languages; they are parsed by the Web server (or by a program integrated with the Web server) when the Web page is accessed.

The business logic layer is the “mind” of the application. It does all the error checking, all the form field validation, and all of the . . . well, all of the business logic. If you need to calculate sales tax, then that code belongs in the business logic layer. If you need to send an email to the warehouse when a package is to be drop shipped to an address other than the customer’s billing address, then all that logic belongs in the business logic layer.

Because interpreted languages (and, by extension, most Web scripting languages) are interpreted at run-time, they are notoriously slow at performing some types of processing (string comparisons in particular). These are tasks better left to compiled, server-side objects. If you need to perform a large number of repetitive operations as part of your business logic, you might want to consider creating a compiled object to handle the processing.

Finally, there is the persistence layer. After performing all your calculations, you will probably want someplace to store some information. At this point the reader may be inclined to say, “Just stick it in a database already!” or “Just write it to a text file!” As simple an idea as this is, it’s always a good idea to abstract the job of storing and retrieving your information. In the event that you need to change databases, for example, you can change your persistence layer code without affecting your business logic. Some scripting languages will make it easy to abstract the database interactions. There are many specialized database solutions in the world today, and many have proprietary enhancements. Choosing an ODBC-compliant database server can ease the pain of transitioning from one database to another.

LAB 8.2 EXERCISES

8.2.1 UNDERSTAND AND EXPLAIN THE PRESENTATION LAYER

- a) Can you identify any scripting technologies that lend themselves to use in the presentation layer?

- b) Imagine that you've just spent three months developing a Web site for a client who likes lime green. Against all your better judgment, you've written a complete site in lime green. Two days before the site is due to launch, the client has a change of heart and wants the site to be purple instead. What do you do and why?

**LAB
8.2****8.2.2 UNDERSTAND AND EXPLAIN THE BUSINESS LOGIC LAYER**

- a) You need to parse a list of phone numbers and extract the area code from within the parentheses. Is this operation best done by script or by a compiled object? Why?

- b) You need to take product information from the database, format the price so it appears in the format "\$00.00," and display it on the product catalog page. Is this operation best done by script or by a compiled object? Why?

- c) Can you identify any technologies that can be used to create compiled server-side objects?

- d) Can you identify any technologies that could be used for interpreted (non-compiled, i.e., scripted) business logic?

8.2.3 UNDERSTAND AND EXPLAIN THE PERSISTENCE LAYER

Often a database will support an ODBC (Open Database Connectivity) interface as well as a native interface. The native interface often provides faster access and more features than the database's ODBC interface.

AB
.2

- a) Why would anyone want to use ODBC when a database supports faster native drivers with more features?
-
-

LAB 8.2 EXERCISE ANSWERS**8.2.1 ANSWERS**

- a) Can you identify any scripting technologies that lend themselves to use in the persistence layer?

Answer: *HTML, DHTML, CSS, JavaScript, Cold Fusion, ASP, PHP, SSI, and even APIs.*

- b) Imagine that you've just spent three months developing a Web site for a client who likes lime green. Against all your better judgment, you've written the complete site in lime green. Two days before the site is due to launch, the client has a change of heart and wants the site to be purple instead. What do you do and why?

Answer: *If you answered "cry," then you've probably been in this situation before. If you suggested running a global search-and-replace on the Web site directory, changing all occurrences of "#00FF00" to "#993399," then you're getting warmer. If you suggested abstracting the site's color process into a server-side include file or variable, then you're on the right track. By replacing the literal string "#00FF00" with a server-side variable, we can change the whole site's colors as often as we like, painlessly. Alternatively, we could use multiple stylesheets, or a single stylesheet and client-side JavaScript to accomplish the same goal, tailoring the entire site to each individual user's preferences. Either way, everything can appear in the user's favorite color scheme, whether that is black and white, reds and blues, or forest green on kelly green.*

The persistence layer can be an extremely versatile tool. Don't just think of it as the layer that does colors and formatting, but remember that it can be as dynamic as the content you're serving. You can pass dynamic content to Java Applets, to Macromedia

Flash or Shockwave movies, to XML (eXtensible Markup Language, the emerging standard for information transfer), to WML (Wireless Markup Language, a standard for cell phone Web browsers), a text-to-speech engine, or you can render some good old-fashioned HTML. The possibilities are nearly limitless.

8.2.2 ANSWERS

LAB 8.2

- a) You need to parse a list of phone numbers and extract the area code from within the parentheses. Is this operation best done by script or by a compiled object? Why?

Answer: Using a compiled script is preferable in situations like this. Implementing processor-intensive or repetitious tasks with compiled objects can substantially increase the speed of the operations performed. Scripts can be 10 to 100 times slower than pre-compiled objects, especially when it comes to repetitive loops and string manipulation.

If your aim is to build a solution that will scale from humble beginnings to taking ridiculous numbers of hits per day, it's a good idea to separate the business logic into compiled objects from the beginning.

- b) You need to take product information from the database, format the price so it appears in the format "\$00.00," and display it on the product catalog page. Is this operation best done by script or by a compiled object? Why?

Answer: This is more easily done with an interpreted script than by a compiled object. Combining presentation and business logic is fairly common, since there are rarely clear lines drawn between a client's requirements for functionality and requirements for appearance.

Scripting the business logic in the page (or into an included file) is generally quicker than compiling server-side objects. It's also great for proof of concept Web sites. There is a caveat, however: Try to avoid mixing display code with business logic. It can be extremely time-consuming to change the page's appearance when the display logic is nested among the business logic.

- c) Can you identify any technologies that can be used to create compiled-server side objects?

Answer: Server-side compiled components can be made with any of your favorite programming languages: Java, C, C++, Visual Basic, VJ++, Pascal, or most anything else. You can build standalone executables, or build yourself some Java Beans or DLLs (on Windows boxes), or you might choose to architect a COM, DCOM, or CORBA solution.

250 Lab 8.2: 3-Tiered Web Application Design

- d) Can you identify any technologies that could be used for noncompiled business logic?

Answer: Cold Fusion (which runs under the Cold Fusion Server), ASP (Active Server Pages, which run under Microsoft's Internet Information Server), JSP (Java Server Pages), or PHP (a recursive acronym which stands for "PHP Hypertext Preprocessor").

We will learn more about these technologies in the next chapter.

AB
.2

8.2.3 ANSWERS

- a) Why would anyone want to use ODBC when a database supports faster native drivers with more features?

Answer: Writing a Web application that is coded to a database's native driver is fine, assuming there will never come a time when you need to switch to another database server. It is extremely painful to switch from one db to another when there is a substantial amount of code in place in a proprietary—and now useless—format.

LAB 8.2 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

- 1) Name the three tiers of application design.
 - a) _____ Web portal, co-branding, e-commerce
 - b) _____ Presentation, venture capital, perseverance
 - c) _____ Presentation, business, persistence
 - d) _____ Perpetrate, design, polish
- 2) The graphical look-and-feel aspects of the Web site belongs to which tier?
 - a) _____ Presentation
 - b) _____ Business
 - c) _____ Persistence
- 3) In which tier should you calculate sales tax for an order?
 - a) _____ Presentation
 - b) _____ Business
 - c) _____ Persistence

4) Which tier is most closely tied to the database?

- a) ☐ Presentation
- b) ☐ Business
- c) ☐ Persistence

5) Which tier would retrieve a list of names from a corporate roster, order them according to seniority, and print out their names in descending size order? Check all that apply.

- a) ☐ Presentation
- b) ☐ Business
- c) ☐ Persistence

Quiz answers appear in the Appendix, Section 8.2.

LAB
8.2

CHAPTER 8

TEST YOUR THINKING

- 1) If you are currently planning a Web application, examine your application's structure and functionality in terms of the presentation layer, business logic, and persistence layer. If you don't currently have a Web application plan lying around, pick something simple—like an online bookstore—and follow along.
 - What is your target browser?
 - Do you anticipate needing to use any client-side code?
 - How do customers navigate the site?
 - Do you need to provide your customers with dynamic content?
 - Do you need to get information from your visitors? Detailed information?
 - How much information will you need to store?
 - What is the most processor-intensive part of your application?
 - What is the most Web server-intensive?
- 2) By examining the URL and URL parameters of a Web site, you can usually deduce how a site provides dynamic content. Active Server Pages have an ".asp" extension, while Java Server Pages end in ".jsp." Cold Fusion pages end in ".cfm." Pages written in CGI generally have a ".cgi" extension. Go onto the Internet and see what server-side programming techniques some of the major sites are using.
 - a) Go to live.altavista.com and read any news story on the site. Can you tell, just from the URL, what kind of server-side code the site is running? Does that tell you what operating system the Web server is running?
 - b) Now check out www.microsoft.com. Click a few links. See if you can identify the server-side technology used to provide content there.
 - c) Surf on to www.casio.com. Do the file extensions there give you any hints as to what technology that site is built with?
 - d) Now visit the San Diego Zoo at www.sandiegozoo.org. How does it provide dynamic content?
 - e) To see some really cryptic-looking URLs, go to Kelley Blue Book Web site at www.kbb.com. Click any link and look at the incomprehensible jumble of letters that makes up the page's URL. Now look more closely. Can you spot the one piece of information that gives away what technology Kelley uses?
 - f) Finally, try to develop an appreciation for well-designed content presentation on the Web. Learn from other people's good design.

INTERNET

ENDORSED BY
WOW
World Organization of Webmasters

Supporting Web Servers

LIBRARY OF CONGRESS



0 007 831 810 7

BEBO WHITE

ABOUT THE ADVANCED WEBSITE ARCHITECTURE SERIES

Endorsed by WOW, the World Organization of Webmasters, these fully interactive workbooks—and their companion websites at www.phptr.com/phptrinteractive—give you the core skills you need to manage content, business, and technology on the World Wide Web. They are building blocks in Prentice Hall's comprehensive curriculum for professional webmaster certification.

ABOUT WOW

The World Organization of Webmasters is a professional organization with thousands of members and affiliates worldwide. A non-profit organization, WOW was created to enhance the role and position of those individuals who create, manage, maintain, and market websites. This support is provided through membership in WOW. For more information on becoming a member, please visit www.joinwow.org.

Master the networking fundamentals every Web professional needs to know—*fast!*

This user-friendly, interactive workbook delivers all the networking knowledge you need to succeed as a Web professional! From network architectures to Ethernet components, optimization to streaming media, Web protocols to DSL access, it's here—organized for easy learning and rapid mastery. You'll find up-to-date techniques for maximizing the performance of your Web connection, securing your site, choosing next-generation technologies, and much more. Coverage includes:

-  **Fundamentals:** network characteristics, network models, WANs, internetworks, intranets, and extranets
-  **Architecture:** packet-switched networks, client/server architectures, Ethernet, network components, and more
-  **Protocols:** IP, TCP, UDP, DNS, HTTP, SMTP, MIME, FTP, MAC addresses, and more
-  **Applications:** mail, Web services, FTP, push and pull technologies, and streaming multimedia
-  **Connectivity:** DSL, T-1/T-3, ISDN, wireless networks, and cable modem connectivity
-  **Security:** encryption, SSL, SHTTP, HTTP, SET, firewalls, sniffers, proxy servers, and VPNs
-  **Web Server Support:** Web development, scripting, JavaScript, Jscript, ECMAScript, CGI, server-side APIs, and dynamic content

If you're going to make the most of your connection to the Internet, you need an in-depth, hands-on understanding of networking. With *Supporting Web Servers*, that's exactly what you'll get—faster than you ever imagined.

ABOUT THE AUTHORS

BENAY DARA-ABRAMS is CEO of BrainJolt.com and the designer of the Web-based Online Adaptive Learning Environment.

DREW DARA-ABRAMS is CTO of BrainJolt.com.

TREVOR PEACE has a degree in Computer Science from Carleton College. He is currently working as a computer consultant in the San Francisco Bay area.

BEBO WHITE is a computational physicist and Computing Information Systems Analyst at SLAC. He has written and lectured extensively on Web technology.

\$44.99 U.S./\$68.00 Canada

PRENTICE HALL
Upper Saddle River, NJ 07458
www.phptr.com/phptrinteractive



0 76092 01075 3

ISBN 0-13-085899-4



9 780130 858993



9 0000

LIBRARY OF CONGRESS



0 007 831 810 7