WILEY

TIMELY. PRACTICAL. RELIABLE.

Mastering IBM® WebSphere® Portal

Expert Guidance to Build and Deploy Portal Applications

Ron Ben-Natan
Richard Gornitsky
Tim Hanis
Ori Sasson

Petitioner has added romanettes to pages (i)-(iii). Otherwise, it leaves the original page numbering.

SAP AMERICA, INC.
EXHIBIT 1005

Mastering IBM WebSphere Portal

Ron Ben-Natan
Richard Gornitsky
Ori Sasson
Tim Hanis

WILEY
Wiley Publishing, Inc.

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2004 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 0-7645-3991-4

Manufactured in the United States of America

10 9 8 7 6 5 4 3

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4447, E-Mail: perm-coordinator@wiley.com.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (800) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data

Trademarks: Wiley, the Wiley Publishing logo and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. [Insert any third-party trademarks.] All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Contents

Foreword xvii

Acknowledgments xxi

About the Authors xxiii

Introduction xxv

Part I WebSphere Portal Primer 1

Chapter 1 Introducing WebSphere Portal 3
- What Is a Portal and Why Do We Need Them? 3
- The WebSphere Family 4
  - Foundation and Tools 5
  - Business Integration 6
  - WebSphere Business Portals 7
- What Is WebSphere Portal? 7
  - The User Experience 7
  - The Developer Experience 12
  - The Administrator Experience 14
- WebSphere Portal Architecture 15
- WebSphere Portal Packaging 17
- WebSphere Portal Platforms 19
- What's New in WebSphere 5.0 (including 5.0.2) 21
- Summary 22

Chapter 2 Installing WebSphere Portal on Windows 2000 Server 25
- Getting Going: A Quick Install 25
  - Getting Information on Your Windows Server 27
  - Installing on Your Windows Server 28

Installing with an Existing Web Server and an Existing WebSphere Application Server 36
Installing WebSphere Portal Fixpack 2 39
Summary 41

Chapter 3 Installing WebSphere Portal on Linux 43
Installing on Linux 43
Getting Information on Your Linux Server 44
Installing on Your Linux Server 46
Nongraphic Linux Installation 56
Uninstalling WebSphere Portal on Linux 60
Directory Structure 60
Installing When WebSphere Security Is Enabled 63
Summary 63

Chapter 4 Customizing WebSphere Portal 65
WPSconfig 66
Configuration Templates 67
Using DB2 with WebSphere Portal 69
Configuring the DB2 Server 70
Installing DB2 Client on WebSphere Portal Server 71
Using Oracle and SQL Server with WebSphere Portal 75
Step 1: Create the Databases 75
Step 2: Create the Database Users 75
Configuring WebSphere Portal to Access a Remote Database 78
Summary 84

Chapter 5 Installing WebSphere Portal Tools and Problem Analysis 87
Portal Toolkit 5.0.2 87
Updating WebSphere Studio's WAS Runtime Environment 88
Installing WebSphere Portal Toolkit V5.0.2 MP 89
Updating to WebSphere Portal Test Environment V5.0.2 92
IBM Lotus Workplace Web Content Management Version 1.1 94
Updating the Initialization Parameters 94
Installing the ILWWCM Web Application 94
Updating the ILWWCM Configuration Files 96
Starting the ILWWCM Web Application 96
ILWWCM Administrative Client Setup 96
Installing the WebSphere Portal Content Management Portlet 97
Problem Analysis 98
Tivoli Web Site Analyzer 100
Summary 103

Chapter 6 Migrating to WebSphere Portal Version 5.0 105
Preparing for a Migration 105
Recommended Migration Environment 106

Performing the Migration 107
Setting up the Migration Property Values 108
Migrating Access Controls 109
Migrating Extended User Attributes 111
Migrating Themes, Skins, and Style Sheets 113
Migrating Portlet Applications 114
Migrating Places 116
Migrating Pages 116
Migrating All User Customizations 117
Migrating Credential Vault Slots and Segments 117
Migrating Credential Vault data 117
Migrating Everything at One Time 118
XML Configuration Interface 118
XML Input File 121
XML Configuration Interface Special Properties 122
XML Configuration Interface Transactional Support 122
Changes in XML Configuration Interface for Version 5.0 123
Summary 124

Part II Building and Administering Portals with WPS 125

Chapter 7 Defining Portals and Pages 127
Portals and Pages 127
Logical Structure of a Portal 128
Browsing through the Content Hierarchy 129
Customizing the Portal Structure 131
Adding New Nodes 131
Defining Page Content 133
Editing Node Properties 135
Page Inheritance with Shared Pages 137
Advanced Layout Editing 138
Locks 139
Removing Nodes 139
Creating a New Portal 140
Summary 142

Chapter 8 Tailoring the Portal Web Design 143
Themes and Skins 143
Default Themes and Skins 144
Choosing Themes and Skins 146
Screens 148
Defining Your Own Themes 149
Creating a New Theme 150
The Anatomy of a Theme 152
Customizing Themes 153
Portal JSP Tags 156
Adding Top-Level Links 159

Defining Your Own Skins 161
Creating a New Skin 161
Modifying Styles 162
Summary 162

Chapter 9 WebSphere Portal Personalization 163
WebSphere Personalization 163
Rules-Based Personalization 164
Resources 165
Content Spots 165
Defining Rules 166
Editing Rules 166
Campaigns 171
The Personalization APIs 173
Resource Interface 174
ResourceDomain3 Interface 174
ResourceManager3 Interface 175
Recommendation-Based Personalization 175
Summary 177

Chapter 10 Portal Administration 179
Your Portal Settings 179
Global Settings 179
URL Mappings 180
Custom Unique Names 181
Supported Markups 182
Supported Clients 182
Search Administration 183
Portal Access Control 190
User and Groups 190
Permissions 191
Resource, User, and Group Permissions 192
Roles 192
Credential Vault 195
Web Clipping 196
Portal Analysis 198
Summary 199

Chapter 11 Document and Content Management within WebSphere Portal 201
Document Manager 201
WebSphere Content Management 205
Creating a Site 206
Setting Up Security 206
Creating a Workflow 208
Creating Your Content Template 210

Categorizing the Content 211
Creating Navigators and Menus 212
Page Design 213
Building the Site 214
Adding Content 215
Configuring WebSphere Portal 216
Syndication and Subscription 220
Summary 221

Chapter 12 Adding Collaboration and Search Components to Your Portals 223
Collaborative Features 223
Lotus Notes Integration 224
Microsoft Exchange Integration 226
Lotus Sametime Integration 226
Lotus QuickPlace Integration 228
Microsoft NetMeeting Integration 229
People Finder 229
Collaborative Components 232
Search Capabilities 234
Searching a Collection 235
Managing Taxonomy and Categorizing 236
Extended Search 237
Summary 237

Part III Portlet Development in WebSphere Portal 239

Chapter 13 Extending Portal Functionality: Portlets 241
What Is a Portlet? 241
Portlet Container 243
Page Aggregation 243
Portlet Modes and States 244
Portlet Applications and Portlets 246
Portlet Life Cycle 248
Portlet Configuration Objects 250
PortletConfig 250
PortletApplicationSettings 250
PortletSettings 251
PortletSession 253
Portlet Objects 254
PortletRequest 255
PortletResponse 255
PortletContext 255
PortletLog 255
Portlet Event Model 256
Action Events 257

Message Events 257
Portlet Settings Attributes Events 258
Portlet Application Settings Attributes Events 258
Summary 258

Chapter 14 Portlet Programming Model and API 259
The JSR 168 API 259
Portlet API in WebSphere Portal 260
Portlet API Comparison 260
Portlet Deployment Descriptors 260
Portlet Processing Model 263
Portlet Life Cycle and Configuration Objects 265
Portlet Services and Objects 267
Portlet URI Addressability 270
Portlet Window 273
Developing JSR 168 Portlets for WebSphere Portal 273
Enabling the Portlet Container for JSR 168 273
Which API Should You Use for Your Portlet? 274
Summary 274

Chapter 15 WebSphere Portlet Development Environment 275
An Example Poll Portlet 275
WebSphere Studio 280
Portal Toolkit 280
Portlet Project 281
Portlet Perspective 281
Portal Server Configuration 282
Portlet Preview 282
Building the Portlet Application WAR File 283
Creating the Poll Portlet Project 283
The Portlet Perspective 283
Deployment Descriptor Editor 291
Generated Source Code 294
Summary 295

Chapter 16 Portlet Development 297
Poll Portlet 297
Implementing the Controller 298
Action Classes 308
Implementing the Action Classes 309
Configuration Mode 310
View Mode 313
Edit Mode 314
Implementing the View Classes 315
Implementing the Model Classes 317
Implementing Persistence Classes 320
DatabaseBroker 321

Implementing Utility Classes 321
Summary 321

Chapter 17 Portlet Interactive Debug and JSR 168 Example 323
Poll Portlet Project 323
Portal Server Configuration 324
Portal Debug Mode 328
Portal Trace Logging 330
Generating a Deployable WAR File 332
Deploying Using the Install Administrative Portlet 332
Deploying Using XML Configuration Interface 333
Developing the Poll Portlet with JSR 168 API 333
Create the Poll Project 334
Generated Portlet Code 335
Deployment Descriptors 336
Poll Portlet Class 338
Modifying the Poll Project Source 339
PollPortlet 339
Action Classes 341
Model Objects 341
View Classes 341
User Identifier 342
JSPs 342
Summary 343

Chapter 18 Struts Portlet Framework 345
Creating the New Project 345
Web Deployment Descriptor 348
Portlet Deployment Descriptor 349
Struts Configuration Files 351
Struts Configuration in Portlet View Mode 351
Struts Configuration in Portlet Edit Mode 353
Struts Configuration in Portlet Configure Mode 353
Action Classes 355
Remaining Poll Portlet Implementation 357
Data Beans 357
Persistence Classes 358
Portlet Controller Classes 358
Utilities Classes 358
View Classes 358
JSP Differences 358
Portlet URL Addressability 359
Generating Portlet URIs 360
Which Implementation Should You Use? 361
Summary 361

IV WebSphere Portal within the Enterprise Environment **363**

Chapter 19 Implementing Authentication for Large Enterprises **365**
Enterprise Identity Management 365
LDAP or Database 366
Rules of Thumb for Designing and Maintaining Your LDAP Server 367
Implementing WebSphere Portal Enterprise Identity Management 371
Setting Up Your LDAP Servers 372
Configuring WebSphere Portal for Your LDAP Server 375
Enabling WebSphere Portal to Access Your LDAP Server over SSL 380
Mapping of Member Manager Attributes to Your LDAP Attributes 383
Summary 384

Chapter 20 Integrating Security and Identity Management Tools with WebSphere Portal **385**
Isn't J2EE Security Enough? 386
Third-Party Authentication and SSO Architectures 387
Integrating TAM with WP 388
Installing TAM Support on WP 389
Setting Up SSO 392
Installing SiteMinder Support on WP 397
Using SiteMinder as a Third-Party Authentication Server 398
Using SiteMinder as an External Security Manager 399
Summary 401

Chapter 21 Designing High Availability into Your Portal Server **403**
The Challenges of High Availability 403
Determining a High-Availability Strategy for WebSphere Portal 405
Implementing a Highly Available WebSphere Portal Solution 409
Vertical Scaling with a WebSphere Portal Cluster 410
Horizontal Scaling with a WebSphere Portal Cluster 411
Configuring WebSphere Portal in a Clustered Environment 412
Summary 419

Chapter 22 WebSphere Portal Support for Web Services and Remote Portlets **421**
A Quick Review of Web Services and Remote Portlets 423
Using Remote Portlets in WP 424
Using Web Services as "Back-End" APIs 425
Using the IBM UDDI Business Test Registry 426
Discovering a Web Service 427
Generating a Java Proxy 428

Using WSRP within WP 429
Configuring WP for WSRP 431
Setting Up WSRP Producers 433
Setting Up WSRP Consumers 434
Futures 436
UDDI and tModelKeys 436
Summary 437

Chapter 23 Integrating External Applications with WebSphere Portal 439
Why Discuss Integration? 439
What Options Can You Choose from? 440
Which Tools Are Available to You? 442
Developing Portlets Using Adapters 443
JCA Resource Adapters 443
WebSphere Adapter for mySAP.com and WSADIE 444
Describing the Interface Using WSDL 446
Generating the SAP Service and Using an SAP Proxy 446
Developing Portlets Using EAI Connectors 449
Using the Portal Application Integrator 452
Using the Credential Vault 454
Credential Vault Segments 455
Credential Vault Slots 456
Using the Credential Vault Service API 458
Using the WebSphere Portlet Catalog 459
Web Clipping 463
Web Clipping Authentication Options 464
Web Clipping Options for URL Rewriting 465
Application-Specific Web Embedding 466
Embedding Functional Portals 469
Summary 470

Chapter 24 Supporting Mobile Users 473
Mobile Users 474
Supporting Multiple Markups 476
WebSphere Everyplace Access 479
Supporting Offline Access 479
Supporting Multiple Wireless Environments 482
Transcoding Technology 483
Using XSL for Adapting Content 488
Using Annotators for Adapting Content 490
Transcoding Plug-ins 493
Enhanced Portlets 493
The Application Integrator Portlet 494
Summary 495

Index 497

Foreword

The explosion of access to information and applications, made possible by the Internet and the World Wide Web, has given rise to a new set of challenges for information technology professionals. Portal frameworks such as WebSphere Portal Server provide capabilities and services needed to meet requirements for consistent administration, look and feel, and navigation across all of these sources of content. In addition, portals enable productivity-enhancing services such as single sign-on and role-dependent personalization that bring the right tools and information to the right people in an organization. When combined with tools such as WebSphere Studio Application Developer, used for the development of specialty components (portlets), a portal becomes not only an essential element of any system integration strategy but also a fundamental tool for the development and deployment of new applications.

Software development frameworks evolve out of the observation of programming patterns that frequently recur throughout the industry. Uses of programming techniques such as CGI–bin .exe's, Pearl scripts, and the like have required programmers to re-create or recode basic services such as memory and thread management, security, authentication, connections to database and back-end data sources, and markup generation libraries. Frustration with these repetitive tasks and initiatives like the Java Community Process gave rise to formalized frameworks of services such as J2EE application servers. J2EE application servers are now commonplace and widely deployed in many enterprises.

As enterprises require more and more integration among their application systems and as Web-based user interfaces have become the hub of application access, the J2EE model has been stretched and adapted. As before,

we can observe patterns of use in many enterprises centered around the *aggregation* of content, applications, and services, which makes it easy to create navigational models over the aggregation. Techniques such as JSP-includes were used to assemble larger applications out of pieces. However, these pieces lacked formal specifications for life cycle management and a contract between the components and their container. From these reuse patterns, IBM and others developed the specifications for a reusable component model, *portlets*. Formalization of portlets led directly to the development of management applications for portlets, or portal frameworks, and to a new breed of products to extend the capabilities of J2EE, such as WebSphere Portal (WP).

Portals and their building blocks, portlets, are quickly becoming "norms" for enterprise class application development and deployment environments. As components, portlets encapsulate Web applications in a life cycle and rendering scheme that makes them manageable and aggregatable by portals. The specifications for portlets are becoming codified by industry standards such as the Java Community JSR 168 and the corresponding Web services standard, WSRP. Portlets can be arbitrarily combined with other portlets to create more complex assemblies and navigational structures. This composition makes possible administrative assembly of applications. As the standards mature, they are being enhanced with capabilities that make it possible for portlets to interact with one another and send data and control signals that make portlets work together. Indeed, portlet composition can be thought of as a generalized component assembly programming model in which individually developed components are wired together to form new, more complex components. This powerful model will fundamentally change the way Enterprise Applications are built. Instead of monolithic application structures, applications will be built of modular, reusable components. Standards will ensure that the development and deployment models are portable and reusable across many environments. In addition, catalogs of prebuilt application pieces will enable business professionals with a keen understanding of the business to construct solutions that today projects require in the IT department.

Looking to the future, we see the continued evolution of the portal-based programming model leading to advanced "workplace" style applications. Workplaces bring an unprecedented level of user and administrator control to the design and deployment of component-based solutions. In a workplace, the role of the individual and the intersection of this role with the work processes that go on in all businesses are paramount. Workplaces, which are fundamentally based on the concepts and technology of portals and portlets, combine advanced notions of community, self-service, business process modeling, and management to form powerful and enterprise productivity-enhancing systems. Mastery of the portal concepts and

techniques will help all information technology professionals, not just Java programmers, to solve the future information and application integration challenges of their organizations.

Ron Ben-Natan, Richard Gornitsky, Tim Hanis, and Ori Sasson have authored an authoritative "must-have" guide to the capabilities of WP. *Mastering IBM WebSphere Portal* contains all that one must know to get started and become proficient with the portal application development and delivery environment. Beginning with the basics of installation and configuration, moving through the user interface concepts of skins, themes and tag libraries, and ultimately system administration, you will come to understand all of the essentials for working with WP. In addition, the authors explore the prebuilt capabilities for collaboration among portal users. Finally, you'll learn the ins and outs of the portal and portlet programming models and best practices for developing and debugging your own portlets.

Portals extend Enterprise Application development to the next level. All application developers and system administrators can benefit from a thorough understanding of the power and flexibility offered by this new breed of system framework. I hope you enjoy *Mastering IBM WebSphere Portal*, and find that the next-generation environment of the WP truly enables enhanced productivity and improved business results.

Douglass Wilson
Distinguished Engineer, CTO,
IBM Lotus Software Division

Acknowledgments

The authors would like to thank the following people whose efforts were critical to the success of this book. Specifically we would like to thank Mike Rhodin, Larry Bowden, Douglass Wilson, Hershel Harris, Brandon Smith, and Bill Swatling for their support and encouragement. We would also like to thank Mike Durham, Kathy Sitar, and Uwe Zimmerman for their help in involving us in the WebSphere Portal 5.0 Beta program. Special thanks go to Theresa Smit, Jeffrey Hay, Stefan Hepper, Jim Bonanno, Lisa Tomita, Rob Davis, Ashok Iyengar, George Fridrich, and Venkata Gadepalli for their invaluable help in performing the technical review.

This book would not have come to fruition without the patience and dedication of our editors Jim Minatel and Scott Amerman.

About the Authors

Ron Ben-Natan is Chief Technical Officer at Guardium, a leader in database and data access security. Ron is an expert in the field of portals, portal integration, and portal security. He has developed portal platforms for the telecommunication and energy industries, a security portal, and has been involved in the implementation of numerous enterprise portals. He is also an expert in distributed computing, J2EE applications, application and database security, and Web services. He has published 8 technical books including several best-selling WebSphere Application Server books and over 40 technical articles.

Richard Gornitsky is a Consulting I/T Architect for IBM Software Services for Lotus whose expertise is in integrating WebSphere Portal in Fortune 500 firms from concept to production. He has extensive experience in the full life cycle development of high transaction solutions, which includes simultaneously managing multiple large complex application development and infrastructure projects. His industry experience includes finance, insurance, telecommunications, pharmaceutical, software manufacturing, and retail/distribution. Gornitsky is a requested technical speaker.

Tim Hanis is a Senior Software Engineer at IBM in Research Triangle Park, North Carolina. He holds computer science degrees from Penn State University and North Carolina State University. He is the lead developer for WP business portlets and has extensive experience in helping customers solve business problems using WebSphere products. Tim can be reached at hanistt@us.ibm.com.

Ori Sasson Ori Sasson is an independent software consultant operating out of Singapore. He is an expert in Java Enterprise Development with J2EE, business integration, data mining, and systems security. He has authored fourtechnical books and several technical articles.

Introduction

If you are a user of Web applications you are also most likely a user of portals. If you develop, manage, or administer Web applications, then you are probably already building or managing Web portals or have thought that you should get started with portal technologies. If you fall into any one of these categories and your Web technology of choice is IBM's WebSphere then this book is for you.

Web portals are becoming the de facto standard for packaging Web applications, and Web applications are increasingly being developed as "portal plugins" (often called portlets). Regardless of whether the domain is consumer applications or business applications, portals have become the consensus user interface for Web applications. In the consumer world all major Web sites present a portal look and feel: Yahoo and MyYahoo, MSN, and Amazon are just commonly known examples. If you work for a large corporation you probably have some form of corporate portal—offering you various human resources (HR), finance, and corporate communication applications in a portal-like environment.

Why are Web portals so successful? Because they bring together important functions such as integration, presentation, organization, and customizations—functions that are needed in every complex application environment. Why have they succeeded in Web application environments? Because these application environments tend to be highly complex, provide tens, hundreds, and sometimes thousands of disparate applications and serve huge user populations—sometimes many millions of users.

Given this very real need, companies have been offering portal server technologies since the dawn of Web applications. But like other information technology domains, portals too have gone through a maturation process.

Some of the early leaders in this space are no longer leaders and some have completely disappeared. On the other hand, many of the large players including IBM, Microsoft, Oracle, BEA, and SAP offer portal products—products that are mature, complete, and very functional.

It is no secret that IBM has been very aggressive and extremely successful with the WebSphere family of products. The WebSphere Application Server started with no market share and today dominates the J2EE application server landscape world; according to some market reviews it has passed BEA WebLogic to become the market share leader. What is less well known is that the same has been happening in the portal space with WebSphere Portal quickly becoming a leader in the portal server space. In fact, WebSphere Portal has the highest new adoption rate within large corporations and is being used within more and more projects and across all industry verticals.

If you are among the many people involved in these portal projects, this book is for you. While WebSphere Portal is a mature product, it is also a highly complex and more specialized one, which means that there are less available resources for you to rely on. It supports many functions and you can use it to do a great number of things and in many ways. This complexity has the unfortunate side effect that WebSphere Portal is not easy to master. The goal of this book is to ease this pain, that is help you master WebSphere Portal.

Overview of the Book and Its Goals

This book will teach you how to install, use, administer, manage, and implement a WebSphere Portal V.5 environment. It will also teach you how to develop and deploy portlets (implementing your custom functionality) on a WebSphere Portal V.5 infrastructure. The focus is not only on developers, but also on administrators, architects, and managers involved with WebSphere Portal projects. If you are a developer you will make the most of this book if you have a Java development background. However, because the focus is on portlets and the inner functions of the portal server, even people with less-than-perfect Java skills will benefit from the book.

Because portals are normally found in business applications environments with a high degree of complexity, the book also shows you how to address the interaction between the portal and other elements in the enterprise. Portal applications are not islands of functionality—if anything they are the fabric that forms the bridges. Therefore, you will learn not only about the portal as a server but also how it interacts with components such as LDAP servers, Enterprise Applications, mobile devices, and even other (vertical) portals.

How This Book Is Organized

In order to master WebSphere Portal V.5, the book will take you through a series of topics starting with those focused on the portal itself and culminating with topics that discuss how the portal fits in within enterprise environments.

You will first learn how to install and customize the portal, as well as how to migrate existing environments to version 5. Next you will learn how to administer and manage portals, including defining portlets, pages, and user interface properties. You will also learn how to use personalization, collaboration, search, document management, and content management within WebSphere Portal V.5. The third set of chapters of the book discuss portlets and show you how to use the portlet API, how to develop portlets, and how to use various development frameworks that make this an easy task. Finally, the last part of the book focuses on WebSphere Portal V.5 within the enterprise environment and shows you how to address topics such as high availability, security and single-sign-on, identity management, Web services, Enterprise Applications, and mobility.

The book starts with an introduction to the WebSphere product family and a review of what WebSphere Portal is responsible for within the complete product family. You will learn about the three dimensions of WebSphere Portal (WP) experience: the user experience, the developer experience, and the administrator experience. You will then get an overview of the WP architecture and what's new in WP version 5.0.

Chapters 2 and 3 will take you through the installation procedures for WP 5.0. WP 5.0 greatly improves upon the various WebSphere Portal Server versions 4.x in terms of installations. If you are using a Windows platform turn to Chapter 2, and if you are using Linux, turn to Chapter 3. If you are using other Unix platforms you can start with Chapter 3 and make changes to the installation procedures based on the WP InfoCenter.

Chapter 4 continues with installation by teaching you how to use Oracle, DB2, or Microsoft SQL Server as the back-end database for your WP instances. In addition, Chapter 4 takes you through WP 5.0 customization topics. You will learn how to use WPSconfig and various configuration templates, how to configure the databases used by WP, and how to configure WP when using remote access.

In Chapter 5 you will complete the installation topics by learning how to install and use the WebSphere Portal Toolkit and the WebSphere Portal Test Environment. You will also learn how to install and activate the Lotus Workplace Web Content Management modules and Tivoli's Web Site Analyzer. Finally, you will learn how to diagnose and troubleshoot problems that you may encounter.

Chapter 6 is the last chapter in the first part of the book and covers topics relating to migration to WP 5.0. If you are already using WebSphere Portal Server 4.x, this chapter will teach you how to upgrade your system to WP 5.0 smoothly and easily. If you are starting fresh on WP 5.0 you can skip this chapter.

In Chapter 7 you will learn about defining elements within your portal. You will learn how to create portals and define pages and how to customize your portal's structure.

Chapter 8 continues this theme by teaching you how to tailor the portal's user interface. You will learn what themes and skins are and how to define your own custom style for your portal. You will also learn how to take existing styles and modify them rather than building new styles from scratch.

In Chapter 9 you will learn about one of the most important topics supported by WP: personalization. You will learn about the various personalization features within WP 5.0, how to use rule-based personalization, and how the personalization API can help you with advanced personalization.

Chapter 10 focuses on portal administration and reviews all of the main administration tasks that you will be required to do when maintaining your portal. You will learn about the WP settings, permissions, and access control and in each topic you will see how to perform the main administration tasks.

Chapter 11 reviews document management and content management within WP 5.0. Like personalization, these topics are often synonymous with portals and the chapter will walk you through some of the main tasks you may need to perform, such as setting up workflows, creating content templates, and categorizing content.

Chapter 12 goes on to complete the second part of the book by introducing you to collaboration and to search. You will learn how to use collaboration components—whether they are Lotus components or Microsoft components. You will also learn how to implement search within your portal, manage search taxonomies and categorizations, and set up extended search capabilities.

In Chapter 13 you will learn about portlets and the focus turns to the development of custom functionality to be deployed within your portal. You will learn about the portlet container and the portlet life cycle and will see how to use portlet configuration objects, portlet requests, and portlet responses. Finally, you will learn about the portlet event model.

Chapter 14 continues with portlets and focuses on the portlet API. You will learn about the JSR 168 API and the various portlet APIs at your disposal. You will learn about the portlet processing model, URI addressability, and about services and objects.

In Chapter 15 you will see how easy it is to develop portlets using WebSphere Studio and the embedded Portlet Toolkit. The chapter (along with

Chapter 16) takes you through a complete development cycle using an example poll portlet. Chapter 16 continues with the poll portlet and show you how to implement the controller and action classes, and how to build the view classes and take care of the model classes.

Chapter 17 teaches you how to develop the poll portlet using the JSR 168 API—providing you will a full arsenal of development techniques that you can use in your environment. The chapter also teaches you how to debug your code and how to use trace logging as part of the development life cycle.

Chapter 18 completes the development part of the book by teaching you how to use the struts portlet framework. Struts is the de facto Web application framework in Java server environments and it is very likely that you will develop portlets that make use of struts' Model-View-Controller paradigm. You will learn how to build both the Web deployment descriptor and the portlet deployment descriptor and how to set up the struts configuration file. You will then revisit the poll portlet and see how it would look within a struts environment.

Chapter 19 moves on from a development focus to address authentication issues and how you can integrate an LDAP environment into your WP 5.0 servers. You will learn how to configure your portal to connect to your LDAP servers and how to implement enterprise identity management.

Chapter 20 continues with this theme and addresses the larger issue of single-sign-on and identity management when using WP 5.0 with either Netegrity SiteMinder or Tivoli Access Manager. You will learn how to configure WP 5.0 to use each of these platforms and the issues you need to think about when doing such a deployment.

Chapter 21 goes on to provide you with planning and implementation skills that will help you set up WP 5.0 in a high-availability environment. You will learn about WP clusters and how to build clusters supporting either vertical or horizontal scaling. You will learn how to configure a clustered environment and how to deploy portlets in such an environment.

Chapter 22 is all about Web services in the context of WP 5.0 After a very quick review on Web services you will learn about remote portlets. You will then learn about two methods for using Web services within a WP environment and for each method you will see what configuration tasks and what development tasks you may need to do within your portal.

Chapter 23 is one of the longest chapters in the book and addresses the issue of integrating external applications into WP. In this chapter you will learn what integration options are available and for each category you will learn what needs to be done. You will learn about using adapters to integrate back-end systems with portlets, using middleware connectors, and about the Portal Application Integrator. You will also learn how to set up back-end single-sign-on and about the Credential Vault service. You will then

learn about front-end integration techniques such as Web Clipping and Web embedding as an alternate integration strategy.

The last chapter in the book, Chapter 24, shows you how to support users who do not only use PCs and desktop Web browsers but rather use mobile and wireless devices to access functionality deployed through WP. You will learn about Transcoding Technologies and about WebSphere Everyplace Access as an enabler for supporting mobile users.

The following table shows the chapters that will be of most interest to portlet developers, portal administrators, portal implementation specialists and enterprise architects:

ROLE	FOCUS CHAPTERS
Portlet developer	13, 14, 15, 16, 17, 18, 22
Portal administrator	4, 5, 6, 10, 19, 20, 21, 22, 23
Portal implementation specialist	7, 8, 9, 10, 11, 12, 19, 20, 21, 23, 24
Enterprise architect	19, 20, 21, 22, 23, 24

Summary

This book is about mastering WebSphere Portal V.5. If you are just starting with WebSphere Portal, we recommend you read the chapters in sequence, starting with the installation of the portal, through the various tasks you will need to perform, and ending with advanced topics relevant to enterprise deployment. If you are already familiar with WebSphere Portal, feel free to jump directly to the chapter that fancies your interest—we wrote the book in a way that almost all chapters present a topic from start to finish and is almost self-contained. In any case, we hope that you enjoy the book and that the book will help you gain mastery over WebSphere Portal V.5.

PART

I

WebSphere Portal Primer

CHAPTER

1

Introducing WebSphere Portal

What is a portal and why do I need it? What does WebSphere Portal offer that I can't do with custom code? In this chapter, we will answer these questions and introduce you to the WebSphere brand and more specifically to WebSphere Portal. We will show you how WebSphere Portal can help you rapidly develop and deploy applications that will impact your bottom line by introducing you to its features and architecture. At the end of this chapter you will understand why you need a portal and how WebSphere Portal meets your portal requirements.

What Is a Portal and Why Do We Need Them?

When the Internet was first introduced, applications were simple and text based. Graphics were soon introduced, and programmers found out that much of the development effort was shifting to look and feel rather than implementing business function. Then developers felt it would be cool to integrate various other sites into their own; however, this generated a smorgasbord look and feel. Security soon became an issue and each site then required authentication. If the developer had integrated multiple sites into his or her own site, the user would endure the frustrating experience of having to do multiple logins on a single site, each with a different name and password. By the 1990s the Internet was experiencing a massive boom when numerous software companies and IT departments were developing software; however, most of it provided the same functionality. Many groups were designing search engines and collaboration software and IT managers were questioning why they were constantly redesigning the wheel and

why productivity and quality were so low. Then, if there weren't enough headaches, Internet applications were required to be accessible on many different devices with very different attributes such as desktops, mobile phones, and palm pilots. Some devices can handle sophisticated graphics and have lots of visual room while others have limited input and output capabilities (e.g., a mobile phone).

Portals, specifically WebSphere Portal, help today's businesses address the Internet challenges. It helps programmers focus on developing the business functionality by letting the portal manage the look and feel, personalization, content management, and security. It allows users to integrate different applications from disparate locations and enables them to seamlessly exchange information. Each user can create a personalized device-independent "desktop" with a consistent look and feel by using drag and drop components. Much of the functionality that is required for Web experience does not have to be programmed but can be found in the portal library, a collection of "shrink wrap packaged" portlets or portal applications.

The WebSphere Family

WebSphere is a software platform that enables you to develop, deploy, and integrate all types of enterprise-scale e-business applications: from business to business, customer to business, employer to employee, on demand, and even legacy.

WebSphere is made up of three components (Figure 1-1): foundation and tools, reach and user experience or business portals, and business integration.

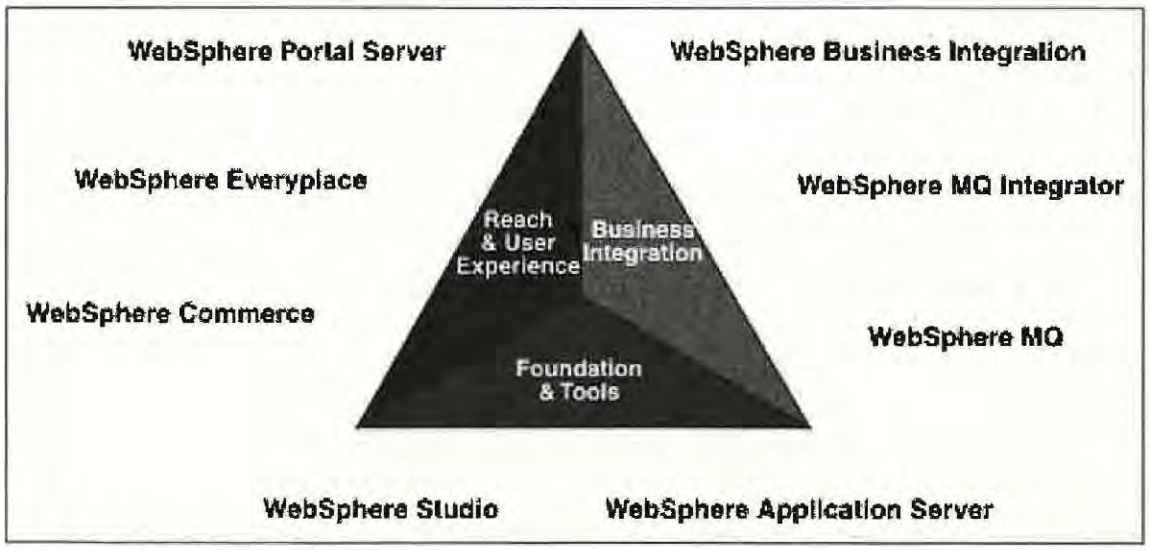

Figure 1-1 IBM WebSphere components.

Foundation and Tools

The "engine" and the "frame" of the WebSphere constitute the WebSphere Application Server (WAS). It is the environment in which you run and manage your Java applications. WAS is a complete, open standard Java 2 Platform, Enterprise Edition (J2EE) 1.4 application server that enables you to perform the following tasks:

1. Integrate easily to other e-business and legacy applications through its support of J2EE Connector Architecture. This gives a consistent way of connecting and communicating to other applications without complex programming.
2. Communicate to other distributed systems using guaranteed and reliable persistent high-performance messaging based on the Java Messaging Service (JMS) and/or Web Services.
3. Deploy and administer applications with ease on a single server. With the optional WebSphere Application Server Network Deployment, you can manage and administer a large number of servers and reduce your administration costs with its automated server management. Installation and administration capabilities are further enhanced through the support of Java Management Extensions, which enables integration with third-party system management products such as Tivoli, Candle, and CA.
4. WAS provides you with performance optimization and reliability by allowing you to control and isolate each application run-time environment. With the WebSphere Application Server Network Deployment option, you get high availability and high transaction support through clustering and caching. Using clustering, it supports load balancing and automatic failover capability. Load balancing can be done at the routers, Web servers, or the application servers. It also provides content-based routing and edge-based caching.
5. Sophisticated and complete security support through
   a. Secure system resources with the Java 2 security model
   b. Standardize authentication with Java Authentication and Authorization Services (JAAS)
   c. Secure communication channels (TLS/SSL) using Java Secure Socket Extension
   d. Security encryption and message authentication using Java Cryptographic Extension
   e. Public Key Infrastructure (PKI) integration based on Java Cryptographic Architecture. PKI manages the issuing,

distribution, and authentication of private and public digital keys. Digital keys are used to authenticate an individual or a resource.

f. Secure interoperability between application servers

g. Support for registries based on Lightweight Directory Access Protocol (LDAP)

h. Single sign-on support using Trust Association Interceptor or Lightweight Third Party Authentication (LTPA)

6. Deploy applications independent of hardware and software environments. WAS can be deployed on Intel, Sun, HP, and IBM hardware platforms running AIX, Linux, zOS, OS/400, Solaris, HP-UX, and Windows 2000/2003.

To enable rapid deployment of applications on WAS, IBM provides WebSphere Studio: a suite of integrated application development tools based on the ECLIPSE open standard framework. From a single user to a large team environment, developers can rapidly develop, debug, and test their WebSphere Applications and then seamlessly deploy into a WAS environment. The WebSphere Studio integrates with the Rational product line including modeling to code generation, testing, and version control.

The foundation and tools component also includes WebSphere Host Integration family, which enables you to integrate your legacy assets with your e-business.

Business Integration

WebSphere Business Integration is a suite of 24 products that provides end-to-end integration by performing the following tasks:

1. Using modeling to design, simulate, and plan business processes
2. Providing products that facilitate linking people, processes, applications, system, and data
3. Enabling you to connect to your customers and partners
4. Allowing you to control and track business processes
5. Enabling you to review, analyze, and improve processes and performance

Items 1, 4, and 5 are supported by the WebSphere Business Integration. Item 2 is supported through message brokering and formatting using WebSphere Business Integration Message Broker. IBM WebSphere Business Integration Message Broker transforms and enhances real-time information between applications that use different message structures and formats.

Item 3 is supported by WebSphere MQ. WebSphere MQ allows you to integrate applications using messaging.

WebSphere Business Portals

WebSphere Business Portal suite focuses on the e-business user experience. It consists of WebSphere Commerce, WebSphere Everyplace, WebSphere Voice, and WebSphere Portal.

WebSphere Commerce enables a user to create selling, buying, and channel management solutions for anything from a simple on-line sales channel to a multitier integrated demand chain.

WebSphere Everyplace provides a suite of tools that facilitate the delivery of Web pages and e-business applications to a broad range of mobile devices.

WebSphere Voice supports development and deployment of conversational and voice recognition e-business solutions. It also provides translation services and unified messaging that includes voice mail, e-mail, and faxes.

The last product, but the most important from this book's perspective, is WebSphere Portal, which, according to IBM's Web site (`www.ibm.com`), provides "a single point of personalized interaction with applications, content, processes, and people." Now we explore this statement and elaborate what it means.

What Is WebSphere Portal?

Previously in this chapter, we discussed what a portal is. WebSphere Portal provides all these capabilities plus the availability and scalability required for large enterprise applications. But to really understand what WebSphere Portal is, you need to look at from the users', programmers', and administrators' perspectives.

The User Experience

WebSphere Portal allows users to create their own virtual desktops that are machine independent. Each user can create his or her own portal pages and customize the content and look and feel of these pages. They can add functionality to their pages through *portlets*, which are portal applications. Portlets are custom made by your sites' programmers, downloaded from the IBM portal catalog, or are available from software vendors. Each portlet has settings that allow users to customize the functionality for their environment. An extensive set of portlets is available from IBM and its partners

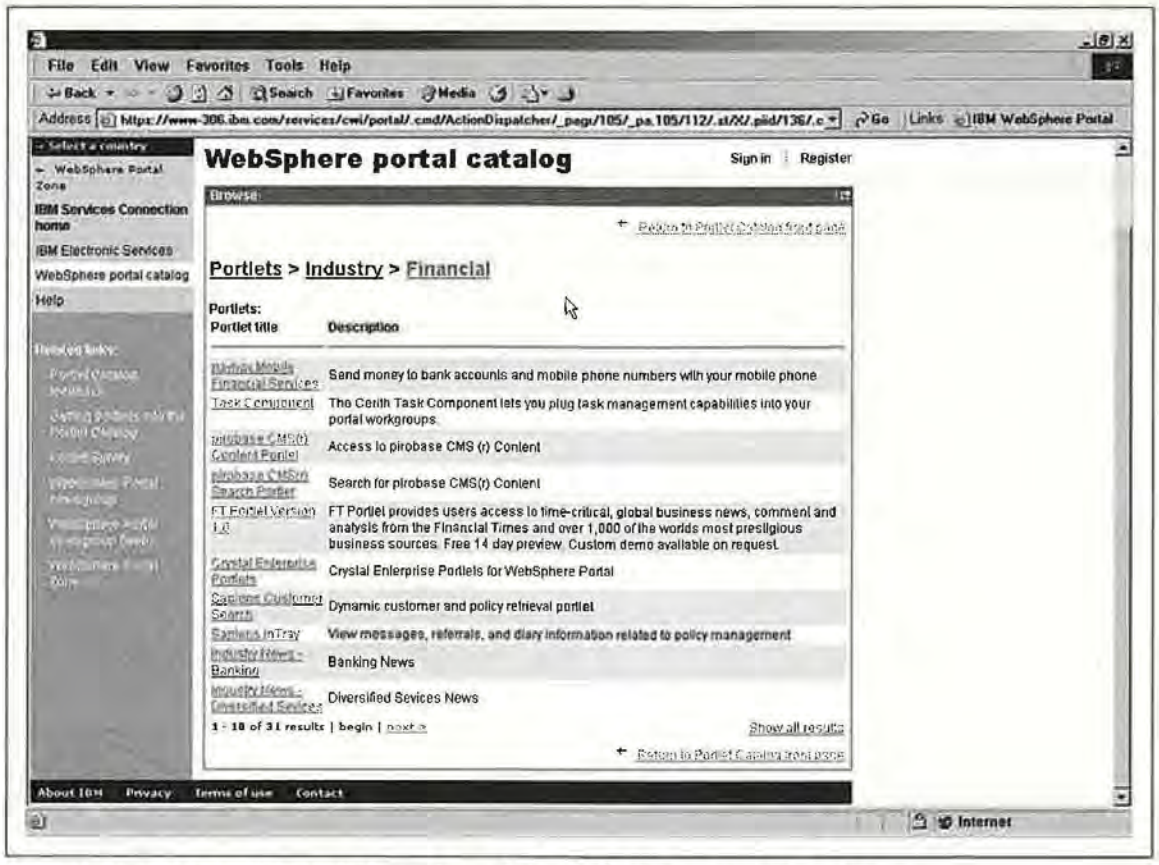

Figure 1-2 Financial portlet section of the WebSphere Portal Catalog.

through the Portal Catalog at `www.ibm.com/software/genservers/portal/portlet/catalog`.

At last count IBM had 531 portlets in its catalog. Figure 1-2 shows some financial portlets available from the catalog. Portlets can be easily added to a page by clicking on the Edit Layout link at the top page you want to customize.

Users can have as many personalized pages as they want. They are arranged in a hierarchical manner with any level of depth starting from their home page. Based on their permissions, users can choose for each page or set of pages their own themes, skins, and layout. Themes define the fonts, colors, spacing, and other look-and-feel components of the page. They are a combination of cascading style sheets, JSP files, and images. Skins comprise title bars, borders, shadows, etc. that surround the portlets. Each page and the pages associated with its tree can have their own theme and skins, thus creating a virtual portal. Each department that enters into the portal will have its own look and feel.

WP Page Navigation supports complex navigation trees and labels that logically group a set of navigation elements.

WebSphere Portal allows users to view content on multiple devices and in multiple languages, including double byte and bidirectional. It also

generates markup that complies with the American Disability Act Section 508 Web Accessibility Standards and the guidelines of the W3C Web Accessibility Initiative. If the portal needs automatic translation, users can incorporate WebSphere Translation Server.

Content Management

WP provides the user the ability to syndicate content or to get up-to-date personalized and filtered content and services from multiple subscriber sources such as Financial Times, Hoovers, Factiva, and others. These subscribers support the Rich Site Summary (RSS) format, so their news and entertainment content can be displayed, managed, and edited with WP's built-in RSS portlet.

WebSphere Portal also enables the user to manage the creation, approval, and publishing of Web content through the support of Web Content Management products. Integration kits are provided on how to publish RSS content from Web Content Management vendors such as Documentum and Vignette. WebSphere Portal also provides its own sophisticated Web Content Management system called IBM Lotus Workplace Web Content Management (ILWWCM).

ILWWCM (formally known as Aptrix) is an enhanced Web Content Management System that replaced WebSphere Content Publishing. ILWWCM is a separately installed component that you need to get from your IBM representative. In the next version, it should be totally integrated into WebSphere Portal.

ILWWCM is designed for organizations with users that publish content to a Web site, whether it is marketing updating product information or human resources updating information on benefits. Content can be published through templates or forms and consists of images, Word documents, Excel, HTML, and so forth.

Through the ILWWCM administrator, users can contribute content, have it categorized, and deployed in a controlled manner using a workflow model. ILWWCM manages the task list of each user and through the establishment of role security enables what content can be changed, created, or deleted by a user. With the use of the workflow model, it coordinates the review and publishing process.

To address more sophisticated content manager requirements, use IBM Content Manager.

Document Management

WebSphere Portal Document Manager (PDM) provides users the ability to view, add, edit, and delete documents within a common folder hierarchy

that can be user-defined. Documents can be made available immediately or to go through an approval process using a defined workflow model. PDM uses WP's access controls, so users can define who can create, read, update, and/or delete a document based on their roles. Authorizations can be set by folder, which will be inherited to the subfolders. PDM provides a subscription capability that enables users to see in their Update folder, for a specified period, a document when it is changed.

PDM supports multiple different types of documents and provides productivity components that enable users to create and edit rich text, spreadsheets, and presentation documents within their browser.

Search

WebSphere Portal provides the user with a vast range of search capabilities. It is built on a sophisticated search engine that supports free-text queries with query assistance and query word completion. Queries can be performed in any language and can use wildcard, advance query operators, synonyms, stop word lists, and fielded search options. The results of the search can be summarized, clustered, and/or categorized using categories that are predefined (over 2,400 categories) or user-defined. WP can index and then search text and 200 other file formats using built-in document filters over the intranet or Internet. Use of the search capabilities is made through the search portlet, the WP crawler, or the document indexer. Both the crawler and the document indexer have a complete scheduler.

Enhanced search is also provided through IBM Lotus Extended Search and DB2 Information Integrator. Portlets using these technologies can integrate and combine other search engines and indexes. Lotus Extended search also allows seamless searches across a collection of Domino servers, databases, and the Internet where the user perceives that the collection of systems is a single server.

Collaboration

WebSphere Portal provides complete support for collaboration through the Notes, Domino, and Domino Web Access portlets. These portlets provide access to collaboration applications that use Lotus Notes databases on Domino servers such as Mail, Calendar, To Do, Notes View, TeamRoom, and Discussion. Additional portlets are available that integrate Domino.doc, Discovery Server, QuickPlace (Lotus Team Workplaces), and Sametime (Lotus Instant Messaging and Web Conferencing).

WP offers Collaboration Center, which is a framework that integrates People Finder, the Web Conference Center, QuickPlace, and Sametime.

QuickPlace provides workspaces for sharing and organizing ideas while Sametime provides instant messaging, shared white boards, and electronic meeting support.

WebSphere Portal enables integration of Domino, Sametime, QuickPlace, and Discovery Server into your portlet by providing Lotus Collaboration Components that are a suite of APIs and JSP tags.

Personalization

WebSphere Personalization server, which is a component of WebSphere Portal, provides the ability to display specific content to site users based on business rules. A business rule is the representation of a business requirement using a coding language based on English language syntax and semantics. The personalization server works by performing the following tasks:

1. Creating a user profile by collecting user information and classifying it into segments
2. Creating models for the content that defines it by its attributes. For instance, a content model might contain product name, price, and age segment.
3. Matching the user segment to the content on the basis of any combination of filtering, rules, or recommendation engine.

The personalization rules are referenced in the portlets. They are created using Web-based tools. You use the rules to classify site visitors into segments and then match the content to the segment.

Another method provided to match site visitors with content is through the recommendation engine. The recommendation engine uses statistical techniques to identify groups of users with similar interests.

WebSphere Portal also provides implicit profiling services that enable the dynamic creation of personalization business rules based on site visitor data.

WebSphere Portal also provides campaign management tools, which enable a business goal to be reached by defining and executing a set of business rules. For instance, you may have a campaign to increase sales of a particular product. To do this, you define business rules that identify likely customers based on the attributes of site visitors or registered users. Content can be displayed on your Web site or personalized e-mail can be sent. Campaigns have start and stop times; they can be prioritized, and can be run in parallel with other campaigns.

Performance

No matter how sophisticated your portal is, it will not be effective if it does not meet your performance, functional, and personalization requirements. WebSphere Portal creates logs that feed into the IBM Tivoli Web Site Analyzer. The Web Site analyzer will analyze the data and report metric in both graphic and text formats so that you can determine the effectiveness of your Web site and focus on areas to improve it.

The Developer Experience

WebSphere Portal allows the developer to focus on functionality development rather than deal with look-and-feel function. Programmers create portlets that are simple Java applications that are extensions to HTTP servlets. Through portlets, they can access and manipulate Web-based content (such as Web pages, syndicated content feeds) and other Java applications/portlets through messaging and/or Web services and not have to deal with issues relating to look and feel. Figure 1-3 shows the page layout and how portlets fit in.

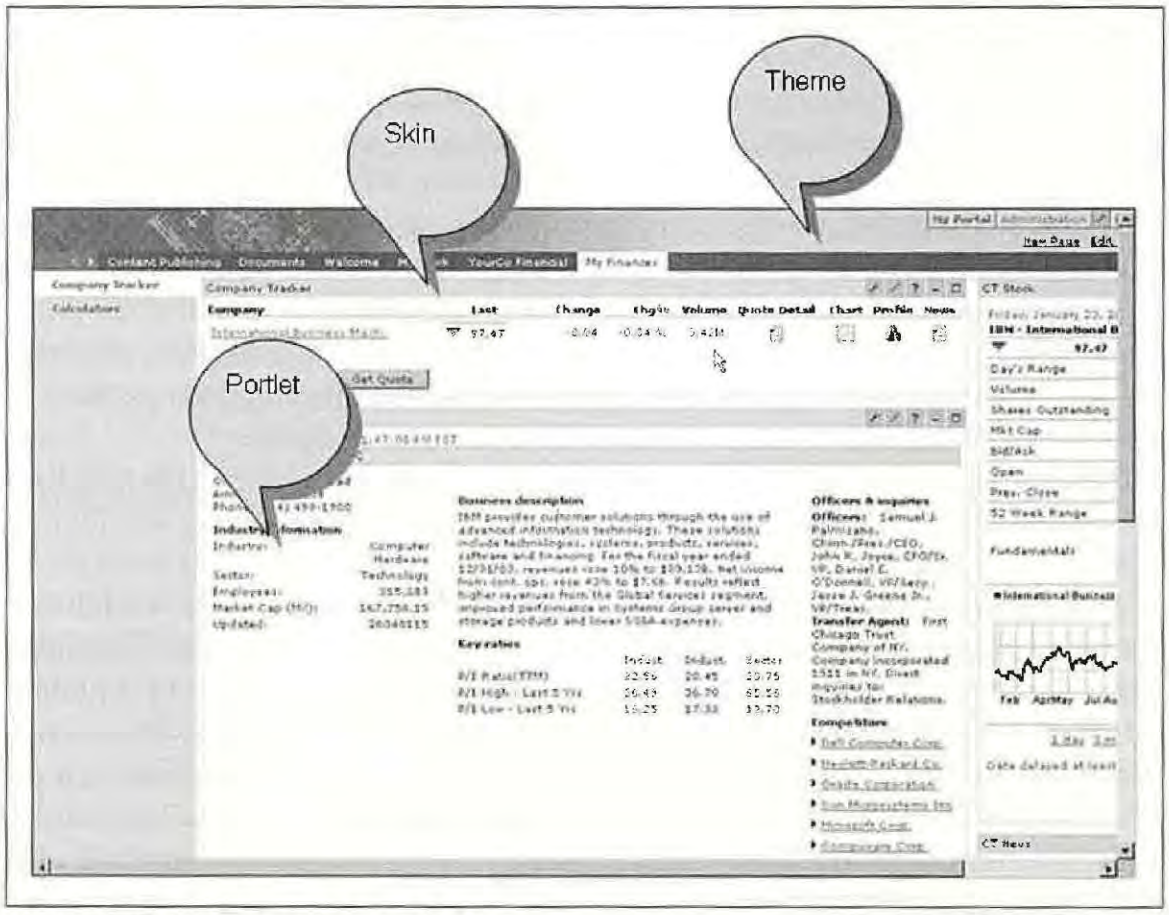

Figure 1-3 The parts of a portal page.

WebSphere Portal also provides a mechanism for portlets to communicate with each other. The developer can send information through a portlet action and a portlet message. For instance, when a portlet displays information on a city, a second portlet can display its weather information by having the city portlet create a portlet action and encode it into the URL that is used for displaying weather. Upon clicking the URL, an action listener is called and then a portlet message is sent to the city portlet for the name of the city.

This method requires an extensive design and coding work. WP provides another method to pass messages independently called cooperative portlets. Cooperative portlets can be "Click-2-Action" or "wired." Portlets express their desire to interact with a broker who at runtime matches data types between the sources of one portlet with the action of another portlet. If a match occurs with a "Click-2-Action portlet," a pop-up menu appears on the page that will enable the user to transfer the data to the connected portlet. If the portlet is wired by an administrator, then the data will be exchanged automatically as opposed to having to wait for the user to click the pop-up menu.

WP gives developers the ability to develop portlets very easily using Web Clipping. This feature enables developers to visually or text select portions of a Web page and create a portlet that displays the information and optionally rewrite the links. Clipping portlet can support other sites' security through the use of WP Credential Vault.

Through the portlet Application Programming Interface (API), WebSphere Portal provides programmers the ability to dynamically discover available services, thus keeping service code independent of the portlet. WP provides discoverable services for managing the portal's content repository, persistent TCP/IP connections, and the Credential Vault. Other services such as mail can be implemented by the portal developer.

WebSphere Portal provides developers with a rich set of security API. Portal server supports the JAAS architecture. JAAS is a component of the standard Java security model that enables authenticating subjects and fine-grain access control. Developers can use the JAAS API to access JAAS-enabled back-end applications and can also use the credential API to authenticate without seeing the credential secret.

WP provides developers with support for Web Services for Remote Portlets (WSRP) 1.0, a new standard that allows "plug-n-play" of Web Services with Portals. WSRP enables remoting the presentation layer as opposed to regular Web services, which focus on remoting application data. WSRP transmits the actual markup rendered by the remote portlet.

WP provides basic support for JSR 168 1.0 portlets. JSR 168 is a Java Specification Request that defines a common set of portlet capabilities and API. This will allow JSR 168 compliant portlets to run in any vendor's portlet container that supports the JSR 168 standard.

If developers need to quickly access Enterprise Application Systems such as SAP, PeopleSoft, Siebel, then they can use WebSphere Portal Application Integrator. This tool allows rapid development of portlets through a GUI. The developer chooses fields and operation that he or she wants enabled in the portlet from objects found by the tool through querying the back-end system. Operations include searching, updating, deleting, and creating records. WPAI can also support integration of relational databases and Domino applications.

Developers can easily develop, test, and debug portlets using WebSphere Studio suite in conjunction with the portal toolkit. Wizards are available to you for creating portlets, building the portlet deployment descriptor, packaging it into a WAR file, and then deploying it into your production or test WP server.

The Administrator Experience

WebSphere Portal enables an administrator to have complete dynamic control over user access right for every portal resource including portlets, pages, and user groups. Control is set by using permissions. The permissions are defined by the user role. WP has defined roles (Administrator, Security Administrator, Delegator, Manager, Editor, Privileged User, and User), which the administrator or user associates with the resource. A role has a set of permissions that determine whether he/she can view, update, delete, or create the resource. If a role is assigned to a page, group, or folder, then resources below the parent inherit the permissions unless they are specifically blocked. A user can also delegate his or her role to other users. Administrators can also lock pages and/or page layout, thus restricting the impact component users can have on the company portal.

WebSphere Portal Member Services provides administration services through portlets for creating, updating, and deleting portal users and for managing their access rights. These services are provided at both the user and group levels. Users can be members of one or more groups and groups can be members of other groups. Users also have the ability to register and manage their own account information in the language of their choice (given that support has been implemented). Portal Administrators can define the information about the user that will be collected. User profile information can be supported in an LDAP and/or a database. Extended information about a user profile that is not supported by the default LDAP schema can be stored in a look-aside database.

Users can administer themselves ("self-care") or a separate administrator can perform the administration. Registration and self-care are done using forms that are easily modified.

WebSphere Portal supports form-based authentication (HTML user prompted form) through WAS or a third-party authentication server such as Tivoli Access Manager or Netegrity Siteminder. Single sign-on is also supported through WAS, and third-party authentication servers are also supported using LTPA tokens. WP also provides a Credential Vault service that allows portlets to store and access securely user IDs and passwords for applications that are accessed on remote systems.

WebSphere Portal gives the administrator the ability to control the portal globally by setting the default language, the cache timeout, handling of new user sessions, taking action when an unauthorized user accesses a portlet, and mapping of URLs.

WP also enables administrators to define new mobile devices and browsers so the portal will know what markup language to use and what the capabilities of the device/browser are.

Of course, WP also provides the standard features that allow users and administrators to easily install, activate, and configure portlets, themes, skins, and pages.

Tools are also provided to ease batch administration, migration, and configuration. Administrators can also enable and control tracing.

WebSphere Portal Architecture

So you now understand what WebSphere Portal does, but you would wonder how it works. WebSphere Portal is basically a suite of J2EE Enterprise Applications running and interacting with each other on the WAS.

In Figure 1-4, you can see the WebSphere Portal Architecture. It consists of three components: the page aggregator, the portlet container and services, and the J2EE engine. It starts with the page aggregator getting an HTTP request from WAS and then performing the following tasks:

1. Determines the device and user information based on the request
2. Collects the pages and portlets requested and then filters them based on the user entitlements
3. The portal dispatches specific Java Server Pages (JSPs), which issue includes for portlet Web modules. These portlet Web modules are invoked as servlets and are then rendered based on their service method (edit vs. view). This process results in the layout of the rows, columns, and graphics of the page.
4. Portlets that perform actions and execute messages such as Click-2-Action are processed first.

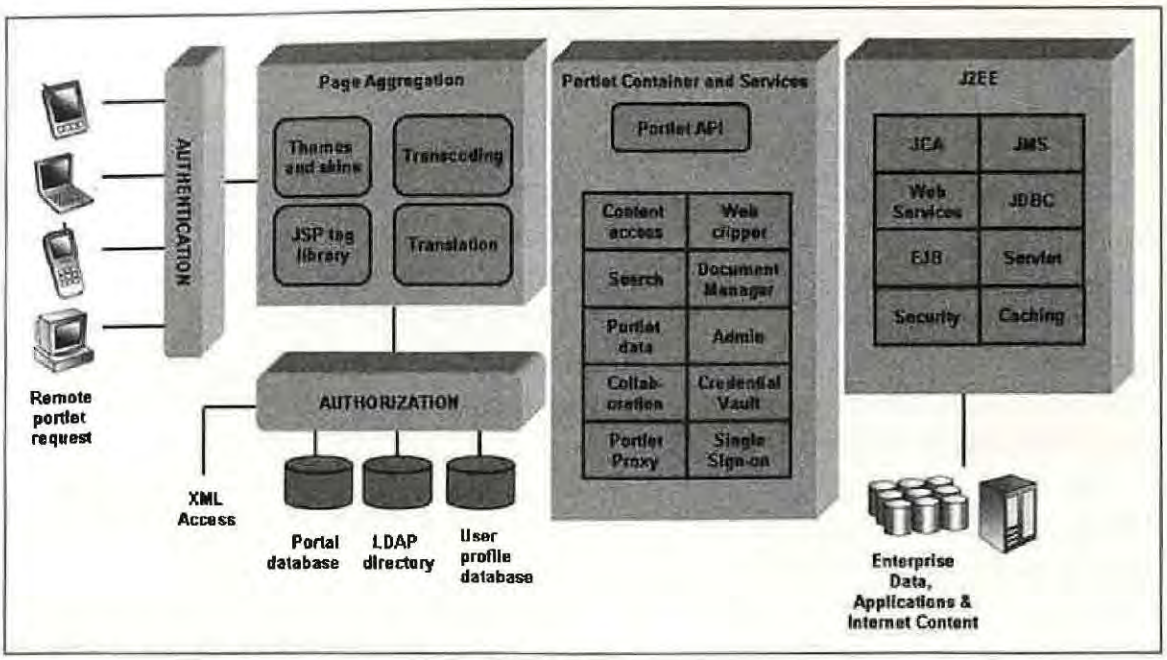

Figure 1-4 The WebSphere Portal Architecture.

5. Portlets that support the target markup are added to the rendering phase.

The second component is the Portlet Container and services. This component controls the portlets and provides services for them to interact with other applications or platforms. Portlets are simple independent HTTP servlets that are reusable and communicate with each other or other resources with well-defined interface. Services, such as Search, collaboration, and Document Manager, extend the functionality of the portlet to support an Enterprise environment.

The standard for the portlet API is JSR 168. JSR 168 is still evolving and as such is still a subset of the functionality of the WP Portlet API. As JSR 168 evolves, the respective WP Portlet API will be deprecated. Presently, both API are very different, have their own containers, and are supported independently.

The last component is the J2EE Engine under which Portal runs. J2EE or Java 2 Platform, Enterprise Edition is an open standard made of components needed to build enterprise applications. J2EE also defines the application and platform model these components run under.

The Java Application Model consists of application components: Java application clients, applets, servlets, JSPs, and Enterprise Java Pages (EJPs). J2EE also defines the Java run-time environment each component runs in. They are called containers and they run on Java 2 Platform, Standard Edition. J2EE also standardizes services which enable applications to integrate with Enterprise Applications. Services comprise JDBC for database

connectivity, JMS for messaging, JavaMail, JAAS for JavaMail, Java API for XML Processing (JAXP), Java IDL, and Java Transaction Architecture for persistent transactions. IBM WebSphere Application Server is fully J2EE certified. Certification is obtained executing successfully the compatibility test suite.

In the WebSphere Portal architecture, you can see that security through authorization and authentication is supported by both the portlet container and services component and the J2EE engine.

WebSphere Portal Packaging

IBM WebSphere Portal V5.0 has two types of packages for the enterprise and two additional packages for small business. Table 1-1 describes the packages for the enterprise. IBM WebSphere Portal Enable for Multiplatforms, version 5.0 is the basic package, while IBM WebSphere Portal Extend for Multiplatforms version 5.0 is the full package.

Table 1-1 Packaging for WebSphere Portal Enable and Extend for Multiplatforms, Version 5.0

COMPONENT	DESCRIPTION	ENABLE	EXT-END
Portal server	J2EE application running in WAS. Provides portal framework consisting of presentation, user management, security, connectivity, and other services. Also includes IBM Cloudscape, version 5.1; a Java-based database.	X	X
WebSphere Application Server Enterprise 5.0 Fix Pack 1	Provides J2EE services for WebSphere Portal. Includes IBM HTTP Server and deployment manager	X	X
IBM Directory Server version 5.1	LDAP Directory Server for storing and retrieving user data required for authentication	X	X
IBM DB2 Universal Database Enterprise Server Edition, version 8.1 with Fix Pack 1	Relational database required to store portal data	X	X

(continued)

Table 1-1 *(continued)*

COMPONENT	DESCRIPTION	ENABLE	EXT-END
Portal toolkit 5.0	WebSphere Studio plug-in that aids in the development, testing, and debugging of portlets	X	X
WebSphere Studio Site Developer version 5.0 with Fix Pack 1	Integrating development environment for building, debugging, testing, and deploying J2EE applications with HTML pages, servlets, and JSPs	X	X
Web Content Management	Provides complete Web content management system, document management, and personalization server	X	X
IBM Tivoli Web Site Analyzer, version 4.5	Provides information on site availability and performance		X
IBM WebSphere Translation server 5.0	Translates Web pages, e-mail messages, and chat conversations into multiple languages	X	X
IBM Lotus Collaborative Components	A suite of APIs and JSP tags that enables integration of Domino, Sametime, QuickPlace, and Discovery Server into your portlet	X	X
IBM Lotus Collaboration Center	Framework that integrates People Finder, the Web Conference Center, QuickPlace and Sametime		X
IBM Lotus Extended Search version 4.0	Enables searches across distributed structured and unstructured data through a single point of access		X
Sametime	Instant messaging and online awareness		X
QuickPlace	Virtual teamrooms		X

Table 1-2 describes the packages for the small business. IBM WebSphere Portal-Express for Multiplatforms, version 5.0 is the basic package while IBM WebSphere Portal-Express Plus for Multiplatforms version 5.0 has some additional features. IBM WebSphere Portal-Express packages come with a simpler installation and the option for user or processor-based licensing.

Table 1-2 Packaging for WebSphere Portal—Express and Express Plus for Multiplatforms, Version 5.0

COMPONENT	DESCRIPTION	EXPRESS	EXPRESS PLUS
Portal server	J2EE application running in WAS. Provides Portal framework consisting of presentation, user management, security, connectivity, and other services. Also includes IBM Cloudscape, version 5.1; a Java-based database	X	X
WebSphere Application Server-Express 5.0 Fix Pack 1	Provides J2EE services for WebSphere Portal. Includes IBM HTTP Server and deployment manager	X	X
IBM Directory Server Express version 5.1	LDAP Directory Server for storing and retrieving user data required for authentication	X	X
Portal toolkit 5.0	WebSphere studio plug-in that aids in the development, testing, and debugging of portlets.	X	X
WebSphere Studio Site Developer version 5.0 with Fix Pack 1	Integrating development environment for building, debugging, testing, and deploying J2EE applications with HTML pages, servlets, and JSPs	X	X
Web Content Management	Provides complete Web content management system, document management, and personalization server	X	X
IBM Lotus Collaborative Components	A suite of APIs and JSP tags that enables integration of Domino, Sametime, QuickPlace, and Discovery Server into your portlet	X	X
IBM Lotus Collaboration Center	Framework that integrates People Finder, the Web Conference Center, QuickPlace and Sametime.		X
Sametime	Instant messaging and online awareness		X
QuickPlace	Virtual teamrooms		X

WebSphere Portal Platforms

WebSphere Portal V5.0 supports the platforms shown in Table 1-3.

Table 1-3 WebSphere Portal Platforms

PLATFORM	NAME
Hardware	▪ IBM-compatible PC with Windows 2000 Server with Service Pack 2 or Service Pack 3, Windows 2000 Advanced Server with Service Pack 2 or Service Pack 3, Red Hat or SuSE operating environment ▪ IBM pSeries with AIX, version 5.1 ML3 or version 5.2 operating environment ▪ Sun processor with Solaris operating environment, version 8 ▪ IBM zSeries or IBM S/390® Parallel Enterprise Server™ capable of running SuSE Linux Enterprise Server, version 7 (31-bit) operating environment
Software	▪ AIX V5.1, 5.2, 5.3 ▪ RedHat Enterprise Linux AS for Intel ▪ RedHat Linux for Intel (×86) 8.0 2.4 ▪ RedHat Linux for zSeries 7.2 2.4 ▪ Solaris 9 (Fix level: 9 12/2002) ▪ Solaris 8 (July 29, 2002 or later) ▪ SuSE Linux for Intel (×86) 7.3 2.4 ▪ SuSE SLES for Intel (×86) 7 2.4 ▪ SuSE SLES for Intel (×86) 8 2.4 ▪ SuSE SLES for S/390 8 2.4 ▪ SuSE SLES for S/390 7 2.4 ▪ Windows 2003 Standard ▪ Windows 2003 Enterprise ▪ Windows 2000 Advanced Server SP2+ ▪ Windows 2000 Server SP2+
Databases	▪ IBM Informix, version 9.3 or version 9.4 ▪ Cloudscape, version 5.1 with Fix Pack 13 ▪ Microsoft SQL Server Enterprise 2000 Service Pack 3 ▪ DB2 Universal Database for z/OS and OS/390, version 7.1 ▪ DB2 Universal Database Enterprise Server Edition, version 8 with Fix Pack 1 ▪ DB2 Universal Database, version 7.2 with Fix Pack 7 and Fix Pack 8

Table 1-3 *(continued)*

PLATFORM	NAME
	▪ Oracle 8i, version 8.1.7 ▪ Oracle 9i, version 9.2.0.1
LDAP Servers	▪ IBM Directory Server, version 4.1 and version 5.1 ▪ Lotus Domino Enterprise Server, version 5.0.12 ▪ Lous Domino, Release 6.0 ▪ Sun ONE (formerly iPlanet) Directory Server 5.0 ▪ Microsoft Active Directory (as included with Windows 2000 Server) ▪ Novell eDirectory 8.6
Web Browsers	Windows platform: ▪ Internet Explorer 5.5 and 6.0 ▪ Opera 6.0, 6.1, and 7.0 Windows, Linux, AIX, Solaris operating environment: ▪ Netscape 6.2 or 7.0 ▪ Mozilla 1.0.2, 1.2.1, and 1.3

What's New in WebSphere 5.0 (including 5.0.2)

WebSphere Portal version 5.0 offers significant improvements over WebSphere Portal version 4.2.1, among which are the following:

- Installation is now simpler and more reliable.
- Cloudscape, a Java database, is installed automatically. Cloudscape requires no configuration and is maintenance free.
- Configuration utility is provided to easily customize WebSphere Porter to use other databases, LDAP Servers, or Web servers.
- Improved database and LDAP server support
- Basic support for WSRP 1.0
- Basic support for JSR 168
- Web Clipping Portlet has been updated to make clipping portlets simpler to create and perform better.
- Search has been greatly enhanced that enables you to find, index, categorize, summarize, and support more than 200 documents. Search will enable you to define your our own category tree or it will automatically categorize using 2,400 predefined categories.

- Cooperative portlets that enable dynamic portlet interaction on a page
- Portal pages can be arranged in a tree hierarchy. Administration is much easier because the child pages can inherit the parent's permissions.
- Application Portlet builders enable you to rapidly create portlets to access and manipulate data from an enterprise application.
- Improved navigation through Portlet menus. Portlet menus can be created using static XML or dynamically generated.
- Ability to map user-friendly URLs to pages so you don't have to use cryptic portal URLs
- A new document repository and management system called PDM. PDM also provides the ability to author documents such as text, word, spreadsheet, and presentations in the portal.
- The administrative interface has been enhanced with new administrative portlets, improvements in the Navigation, new and improved themes and skins, and context-sensitive links.
- Lotus Collaboration Center has been enhanced with portlets for People Finder, support for QuickPlace in My Lotus Team workplaces, support for Sametime in Lotus Web Conferences, and more.
- The business portlets have been enhanced by adding an Internet Mail Box, single Notes and iNotes portlet, a Newsgroup portlet, and Mylist portlet (simple check-off list).
- XML access, which is the batch processing interface for the portal, has been enhanced in function and documentation.
- Improved handling of error messages by enabling error messages to be localized
- Transcoding can now be used at the portlet level by administrators.
- Portal Toolkit has been updated for version 5 and includes new and improved Wizards.
- Improve performance through WAS cluster support, performance instrumentation, reduced session data, unified caching architecture based on Dynacache, smarter invalidation, and faster login times

Summary

In this chapter we introduced you to portals, which is the WebSphere product family, and more importantly to WebSphere Portal. We gave you

grounding on the WP architecture, its packaging, and supported platforms. We showed some great features introduced in this version. Now that we wetted your appetite by telling you at a high level all the great things about the WebSphere Portal, let's we give you some substance. The rest of the book will show you how to install and use the product for an Enterprise environment. We will start with the first steps so you can install the product in various environments and program a simple portlet. After each chapter we will build upon the topic so that by the time you complete reading the book you will be able to deploy WebSphere Portal in an enterprise environment.

In the next chapter we will introduce step 1 to deploying WebSphere Portal: installing a simple portal.

CHAPTER

2

Installing WebSphere Portal on Windows 2000 Server

Before WebSphere Portal version 5.0 came along, users had to dedicate an inordinate amount of effort to figure out how to successfully install WebSphere Portal Server. Installations were complex, took a long time, and required huge number of parameters to input. The installation process was inflexible and intolerant of incorrect parameters to the point where an incorrect value would cause the installation to stall and the installation would have to be manually backed out and redone. Users would feel great pride at the end of the installation process that they were able to complete it, almost like they got a 1600 on their SATs.

IBM clearly recognized that this situation was unacceptable. The company has made a huge effort to simplify the process. With WebSphere Portal Extend, you need only four disks to get started. (You get 57 disks in all, but the rest are to support different platforms and languages. They also include products that you will need as your use of WebSphere Portal becomes more sophisticated and your sites get larger.)

Getting Going: A Quick Install

So you want to install WebSphere Portal quickly on a clean system. Figure 2-1 shows that WebSphere Portal 5.0.2 is going to be installed on one server, with WebSphere security not enabled. This includes the supporting components WebSphere Application Server Enterprise version 5.0 Fix Pack 2 (with some additional fixes), IBM HTTP Server 1.3.26.1, and Cloudscape V5.1.26. Cloudscape (really DB2 Java) is an install-and-forget database. It is used to store information about user identities, credentials, permissions,

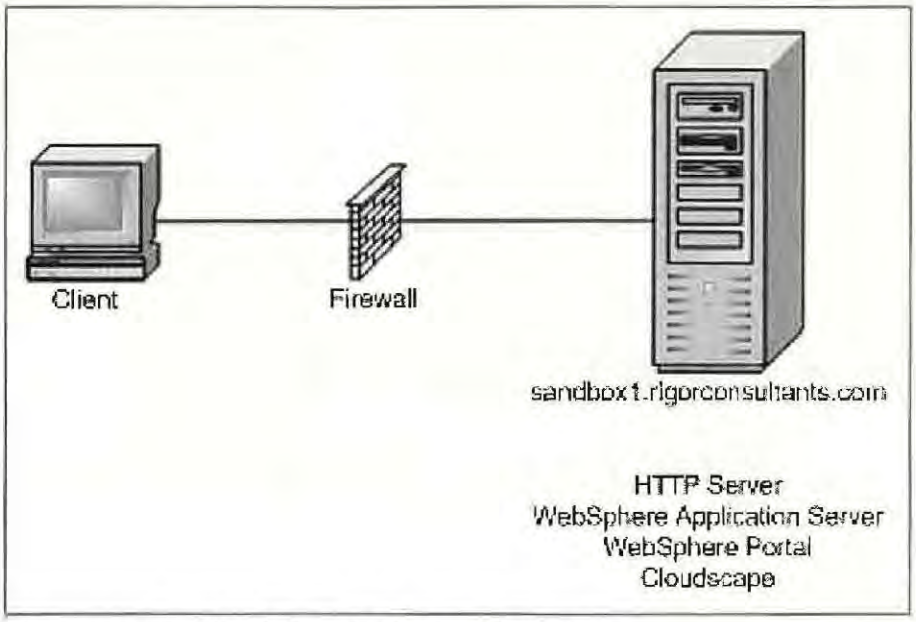

Figure 2-1 A simple installation configuration.

and documents. Designed to be maintenance free and dynamically tuned, it is targeted for sites with less than 2,000 users. It is great for just getting started with small-to-medium-size businesses.

First, you need some hardware, specifically the following:

- Pentium 4 processor at 1.4 GHz or higher
- 1 GB of RAM or more per processor (this is very important)
- At least 3 GB of disk space on an NTFS file system. If you are using Cloudscape, which is the default database, you should have a lot more.
- Network connectivity with a static IP address and a configured fully-qualified host name. WP can work with a dynamic IP configuration if you do not use single sign-on with WAS security enabled. The portal system must be able to resolve an IP address from its fully-qualified host name.

On the software side, you need Windows 2000 Server or Advance Server SP2+. It runs fine on SP4. WP should work with the following browsers:

- Microsoft Internet Explorer 5.5+ on Windows
- Mozilla 1.0.2+ on Windows and Linux with the Java script enabled
- Netscape Communicator 6.2+ on all platforms
- Opera Web Browser 7.11+ on Windows

We really encourage IE 6.0 SP1 to be used for best results.

Getting Information on Your Windows Server

Before you start, you need to perform some tasks on your Windows Server to make it ready for the install. Specifically:

1. Identify your service pack version by typing **winver** in a command prompt. Make sure it is Service Pack 2 or greater.
2. Make sure you have a static IP address by going to Start ⇨ Settings ⇨ Network and Dial-Up connection. Right-click on your network adapter LAN connection and select Properties. Select Internet Protocol (TCP/IP) and click Properties. Ensure that Use the following IP address is selected.
3. Make sure that your server has a fully qualified host name and is properly configured. In a command prompt, ping your server's fully qualified domain name and see if your IP comes up and four successful replies are returned.
4. Make sure that the following ports are not being used by typing **netstat–an** in a command prompt:
   - 443 (standard HTTPS port)
   - 523 (DB2 Administration Server)
   - 8008 (IBM HTTP Server Administration port)
   - 50000 (DB2 instance connection port)
   - 50001 (DB2 instance interrupt port)
   - 50002 (DB2 Control Server)
   - 55555 (WAS database port)
   - 9081 (Default Portal Port)
   - 9090 (WAS Administration port)
5. If you are going to set up WebSphere Application Server or IHS as a service, make sure that the user account you will use to install WebSphere Portal is a local Windows user ID, part of the administrators group, and has the user rights to Act as part of the Operating System and Log on as a service. To add or view User rights select Start ⇨ Programs ⇨ Administrative Tools ⇨ Local Security Policy. Double click Local Policies and then double click User Rights Assignment. Double click on the user right you are interested in and then modify it if required.
6. Make sure Microsoft Internet Information Services is not running since it listens on the same port as IBM HTTP Server. Go into Services under Administrative Tools and stop or disable World Wide Publishing Service.

7. Make sure that PATH environment variable is not set too long. If it is, you may get a message in the `wpsinstalllog.txt` stating the input line is too long or the syntax of the command is incorrect.
8. Disable any firewall products.

Okay, now you are ready to go.

Installing on Your Windows Server

To install WebSphere Portal:

1. Disable any firewall on the server where WP will be installed. After the install, re-enable it and open the appropriate ports and set appropriate rules.
2. Insert the Setup disk in the CD drive. The program should automatically start unless the autostart function is disabled. If it is so, then run `install.bat` from the root directory of the CD.
3. Select the language for the install and click OK.
4. The Welcome panel will then be displayed. You will then be given the option to browse the infocenter if you choose Launch Infocenter. Otherwise click Next.
5. Figure 2-2 will then appear. This is the license agreement. Review the license agreement and if you accept click Next. If you don't then hit Cancel and return the software.
6. The installation then checks the machine for operating system and software prerequisites. If you are installing on Windows XP Professional (yes, you can but it is not supported so users beware!), you will get an error message but just ignore it and hit Next.
7. Figure 2-3 will then appear giving you a choice of a Full or Custom install. Choose Full.
8. In Figure 2-4, you will then be asked to specify the directory to install WebSphere Application Server. If the directory does not exist, it will be created. By default, IBM suggests that you install in `c:\Program Files\WebSphere\AppServer`; however, you may instead want to install in `c:\WebSphere` since the directories are easy to get when you are accessing them in a command prompt. Enter the directory and click Next.
9. The next screen (see Figure 2-5) asks you to specify the directory to install the IBM HTTP Server. Choose the default. If the directory does not exist, it will be created. Click Next.

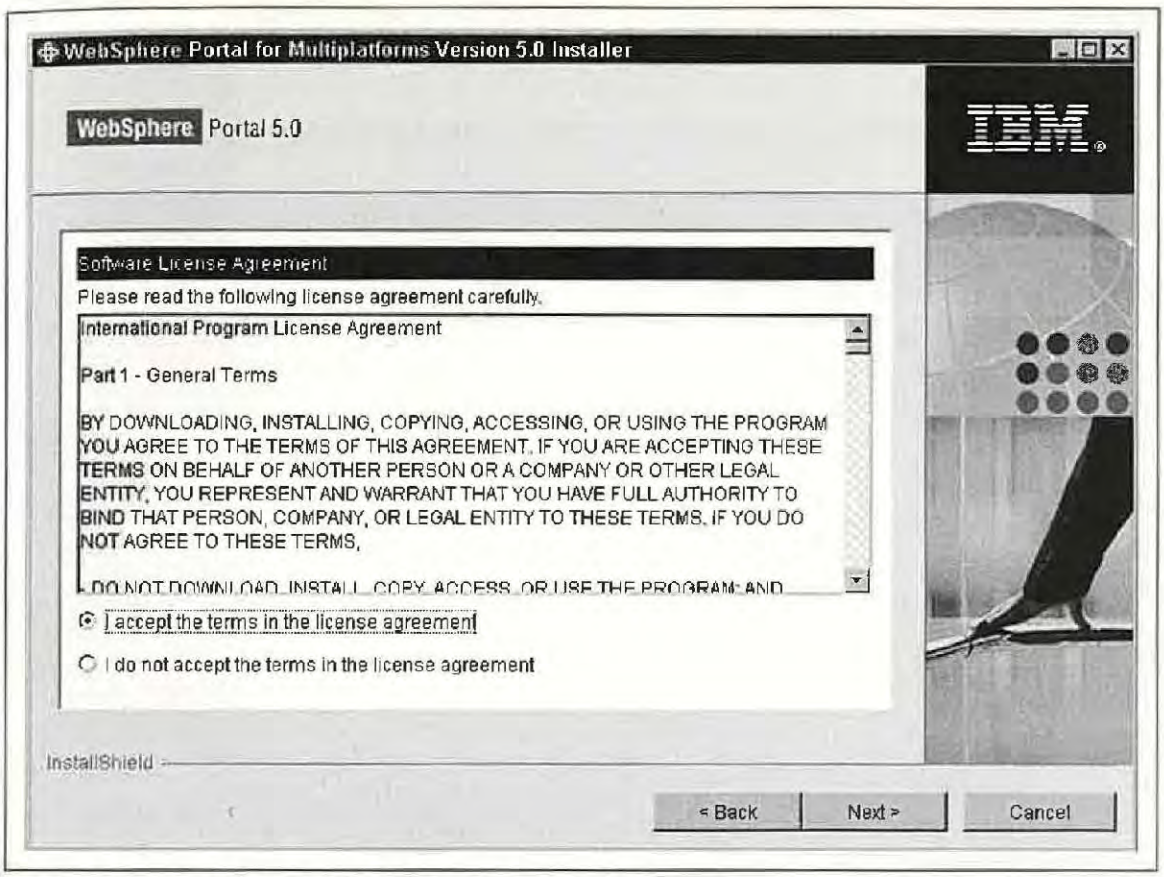

Figure 2-2 WebSphere Portal license agreement.

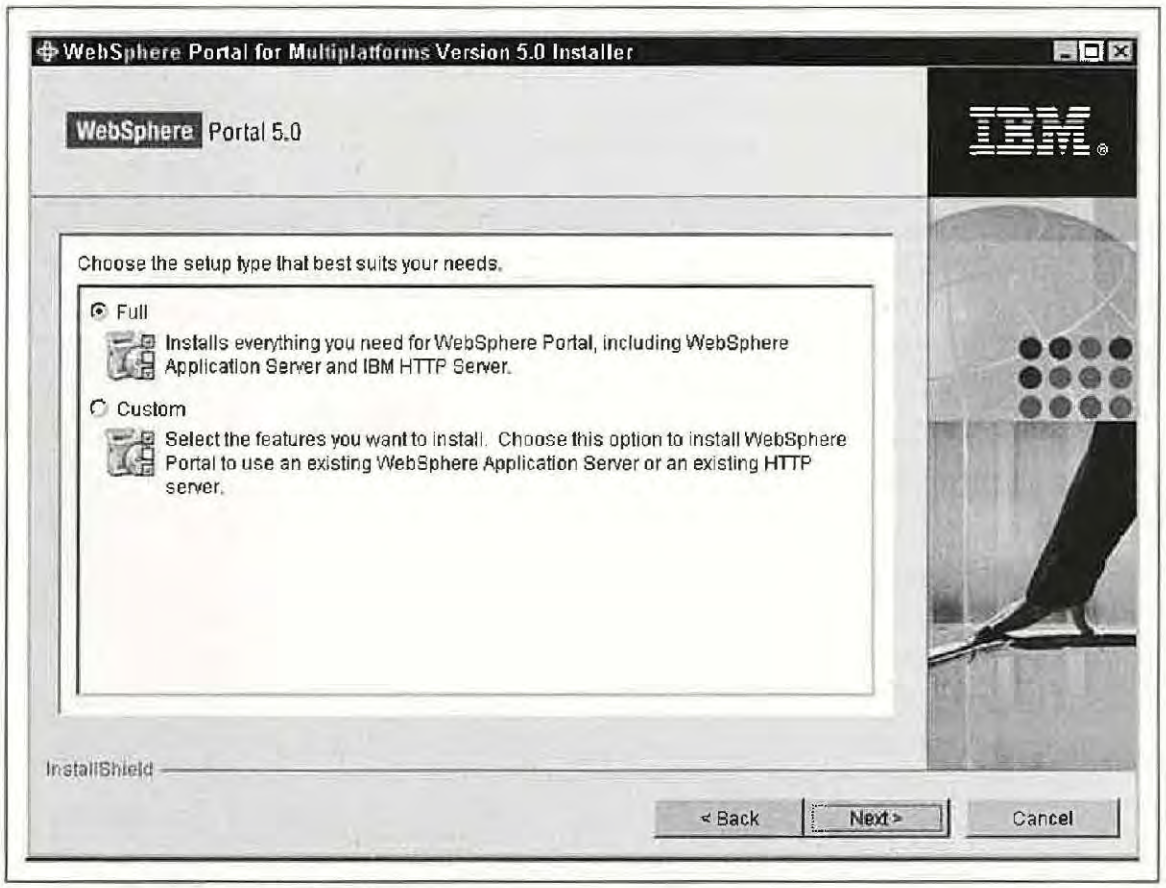

Figure 2-3 Choosing installation type.

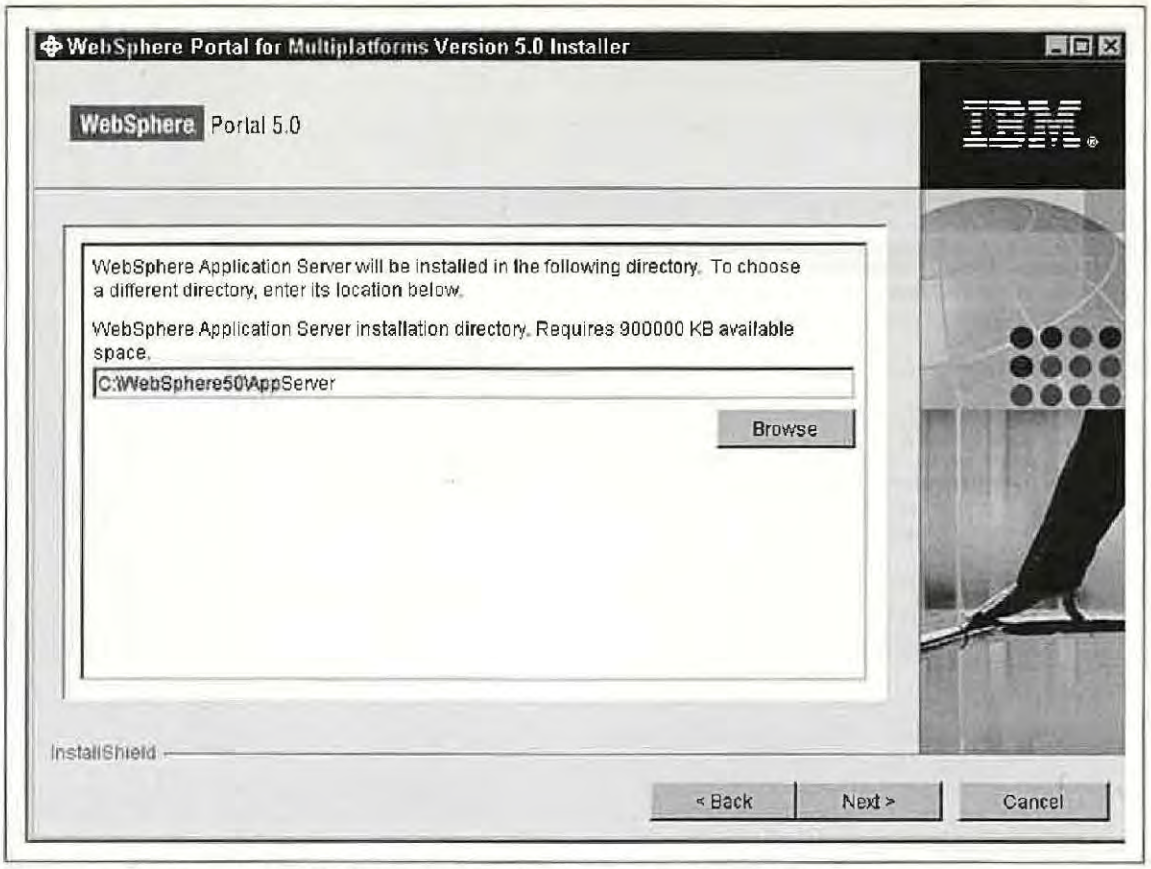

Figure 2-4 Specifying WebSphere Application Server installation directory.

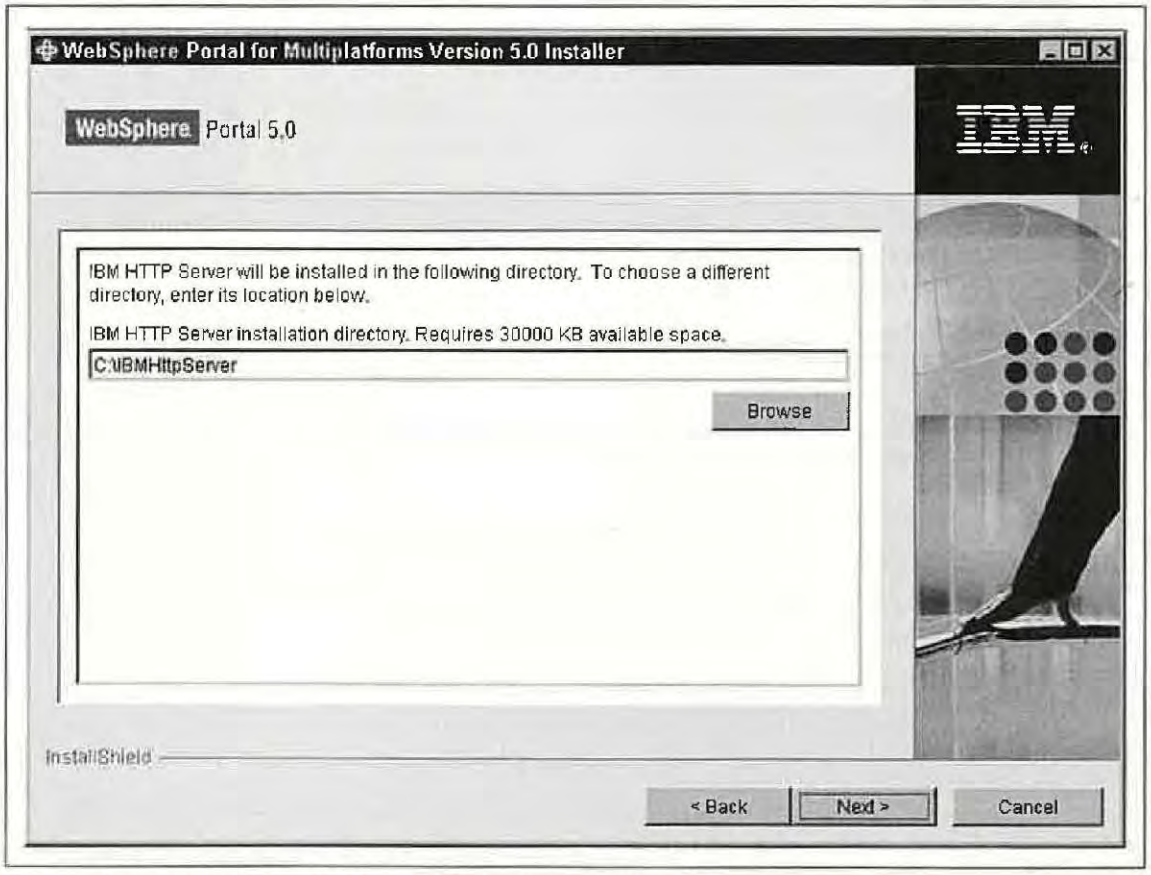

Figure 2-5 Specifying IBM HTTP Server installation directory.

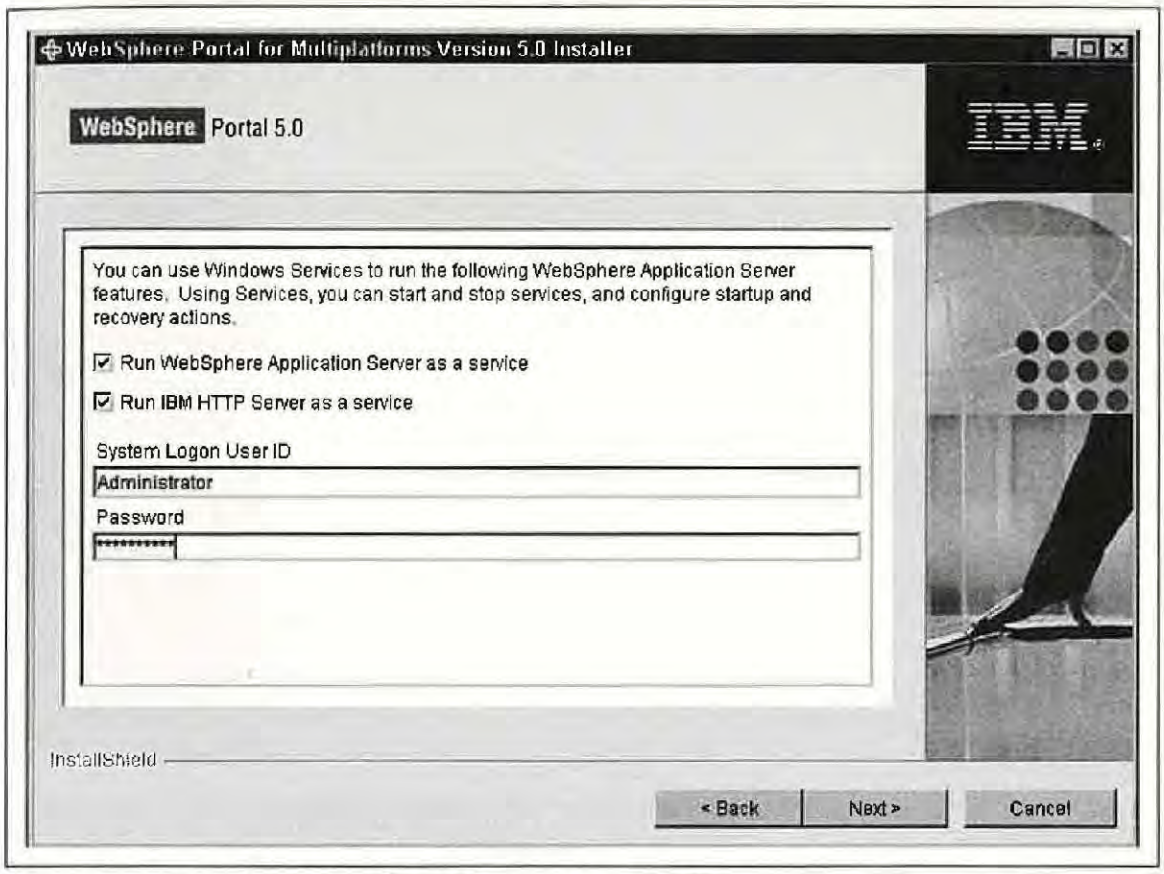

Figure 2-6 Specifying whether you want to run services.

10. Now you are asked (see Figure 2-6) if you want to run WebSphere Application Server and IBM HTTP as services. If you choose this option, specify a user name and password that has administrator privileges. The user ID may have the policy rights to "Log on as a service and Act as port of the operating system." If the user ID does not have them, the installation program will update the user ID and ask you to log out and log back before continuing with the installation. You should install WebSphere Application Server and IBM HTTP as services so they can be run remotely without the administrator logged in.
11. Specify the node name (see Figure 2-7) where you want the WebSphere Portal application to be installed within the WebSphere Application Server cell. This value must be unique and is usually the host name of the computer. Also enter the fully qualified host name or the static IP address of the computer running WebSphere Application Server. Click Next. In Figure 2-6, you choose a node name sandbox1 and a fully qualified host name `sandbox1.rigorconsultants.com`.

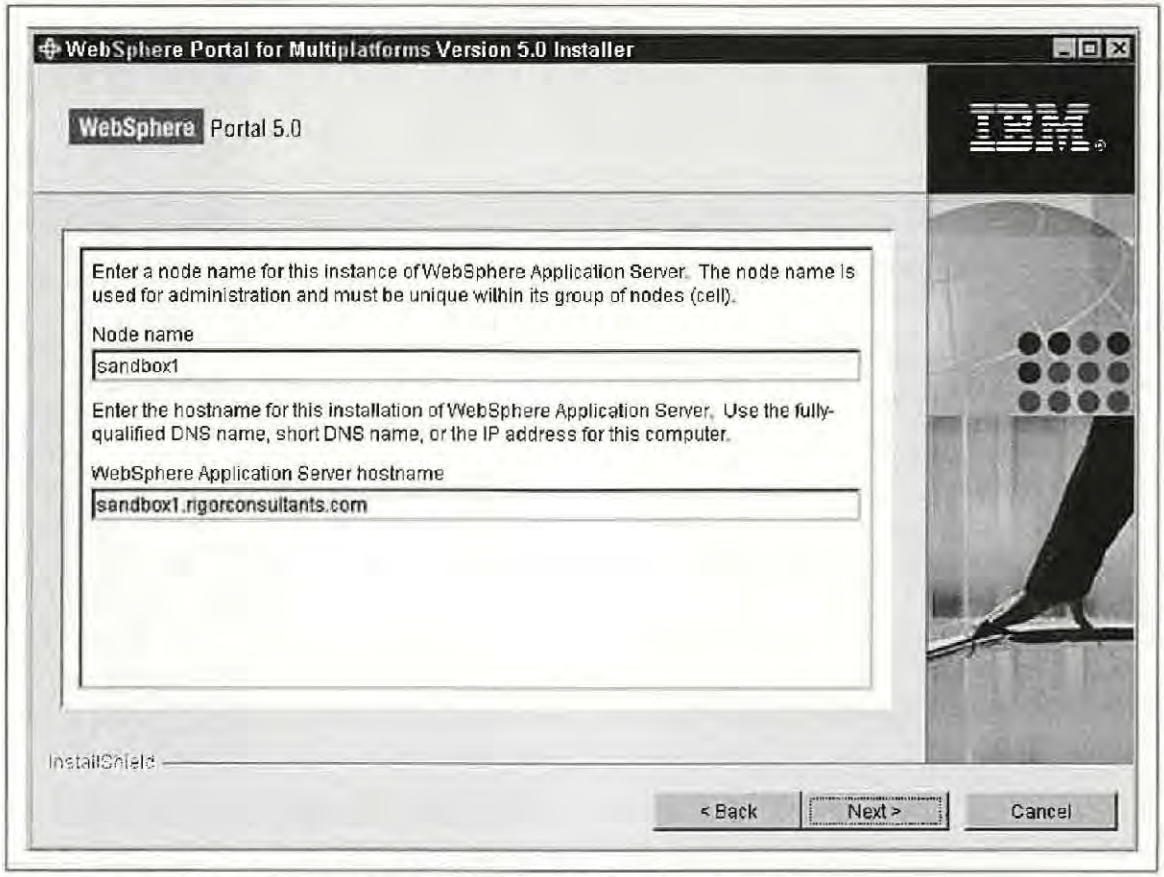

Figure 2-7 Specifying physical location of WebSphere Portal.

12. In the next screen (see Figure 2-8), you will be asked where you want to install WebSphere Portal. Usually it is installed off the WebSphere directory root as a peer to the WebSphere Application Server. Click Next.
13. Enter the user ID and password for the WebSphere Portal Administrator (see Figure 2-9). The password and user ID must be at least five letters long and contain no blanks. This user ID is specific to the WebSphere Portal and does not have to be an Operating System User ID but it must conform to the rules of the LDAP if you are using an LDAP to manage your users.
14. Verify the components to be installed and click Next.

That's all the information you need to give. It's that simple. After you click Next, the program starts installing the selected components. You see the progress indicator moving as the components are installed. It will probably take an hour or two, depending on your system's performance. When the installation is finished, a confirmation panel is displayed with a listing of the components installed. The confirmation panel will also display the port number used to access the portal. The port number is also stored in the

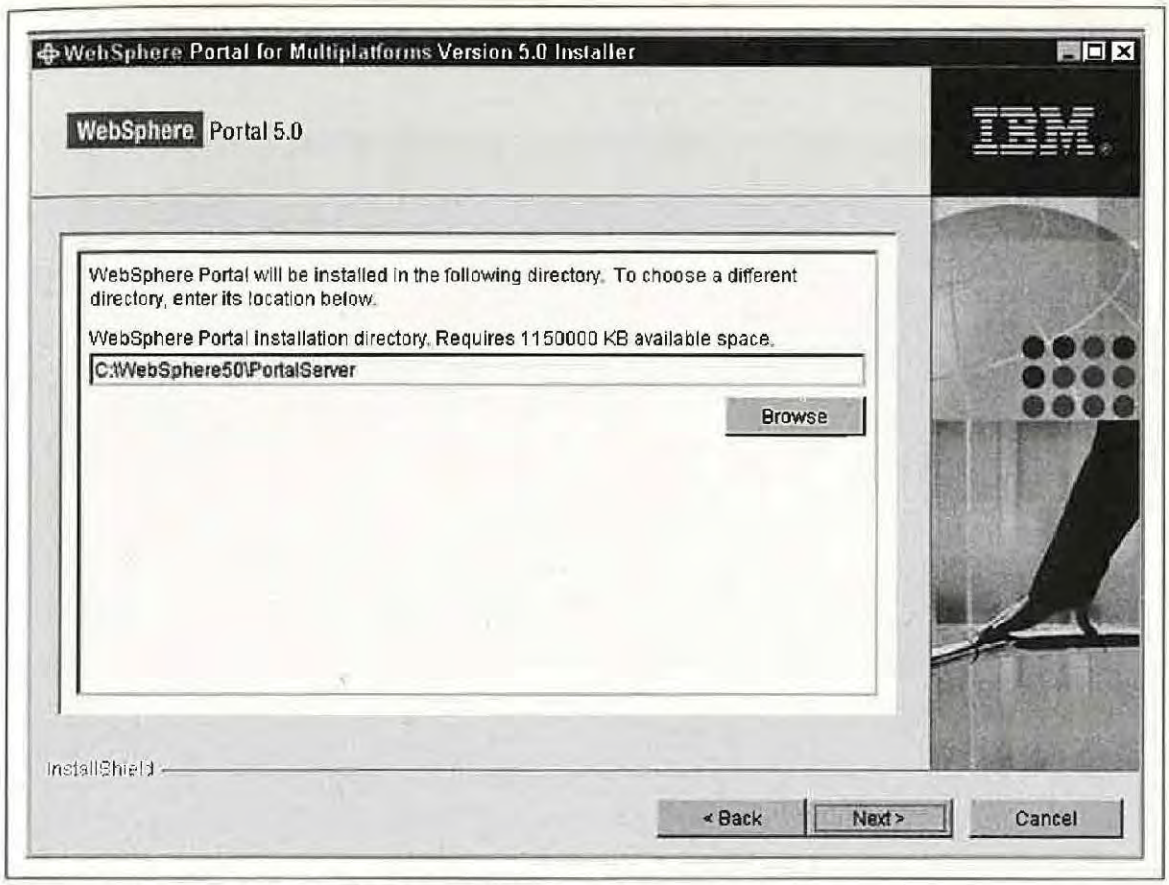

Figure 2-8 Specifying WebSphere Portal installation directory.

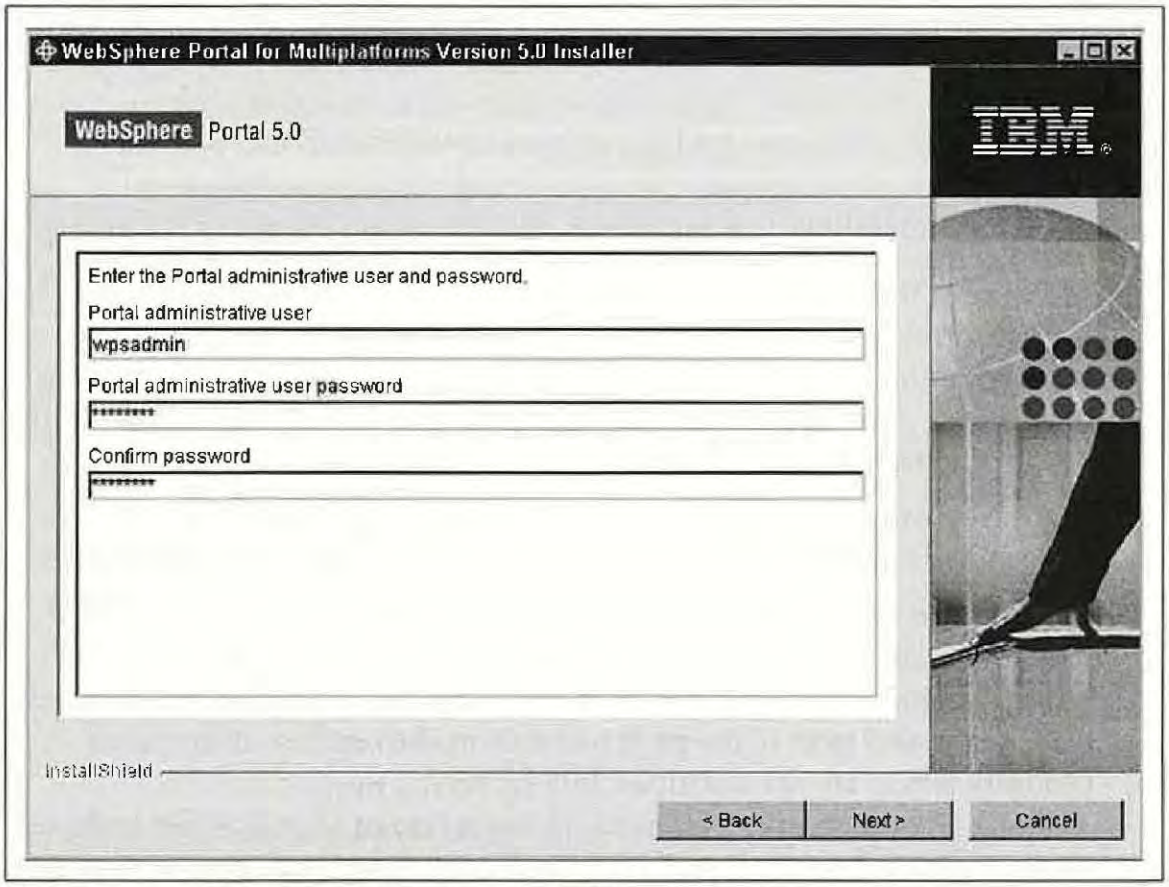

Figure 2-9 Defining WebSphere Portal administrator.

Figure 2-10 Starting the Portal Server.

WpsHostport property `<wp_root>/config/wpconfig.properties` file. Record the property number and click Finish. Menu entries will be added under Start ⇨ Programs for IBM HTTP Server and WebSphere Application Server and WebSphere Portal. Since IBM HTTP was set up as a service, it will start automatically when you boot the server. However, to start the Portal Server you will need to click Start ⇨ Programs ⇨ WebSphere ⇨ Portal Server V5.0 ⇨ Start the Server. A command box will appear. If you have a successful start, you will see entries in the command box similar to Figure 2-10.

To verify the installation was correct, the following steps are necessary:

1. Make sure the IBM HTTP Server is installed and is not in conflict with another HTTP server such as IIS or Apache. On your Web browser enter your fully-qualified host name in the address bar (e.g., `sandbox1.rigorconsultants.com`). You should see a page with "Welcome to IBM HTTP Server" (see Figure 2-11).
2. Start the portal server by going to Start menu and clicking Programs ⇨ IBM WebSphere ⇨ Portal Server V5.0 ⇨ Start the Server.
3. Enter in the address bar in your browser the URL `http://<hostname.yourco.com>:,port>/wps/portal` where hostname.yourco.com is the WebSphere Portal fully-qualified host name and port is the port number in the confirmation panel (usually 9081). In our example. It is `http://www.rigorconsultants.com:9081/wps/portal`. Click Go and you should see the WebSphere Portal Welcome Page as shown in Figure 2-12. Please note that a full installation does not

Figure 2-11 IBM HTTP Server Welcome screen.

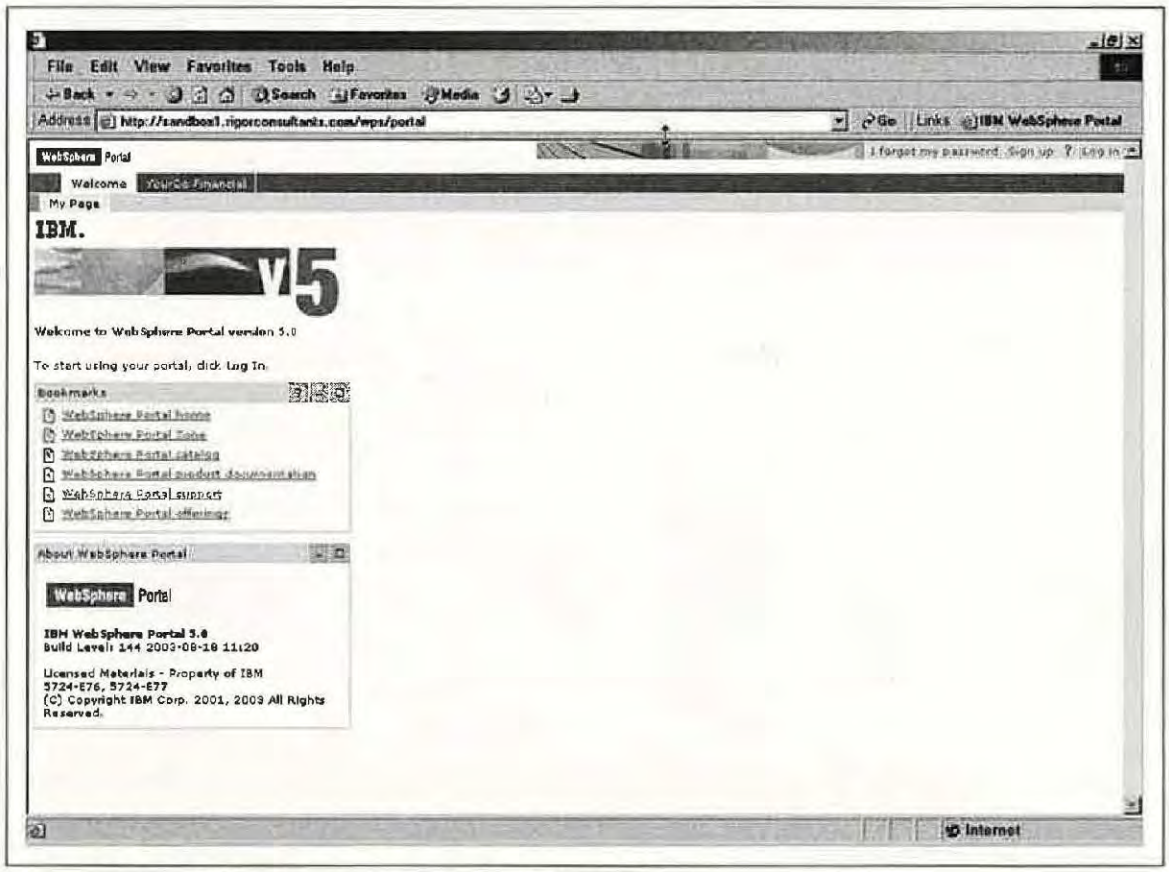

Figure 2-12 WebSphere Portal Server Welcome screen.

configure WebSphere Portal with the IBM HTTP Server even though it is installed with it. This must be done separately. By default, WebSphere Portal uses the WebSphere Application Server internal HTTP server (port 9081).

Installing with an Existing Web Server and an Existing WebSphere Application Server

This installation is relatively easy. Except in the real world, you rarely install with a new instance of the Web server and a new instance of WebSphere Application Server. Here is an example of a more realistic install. Figure 2-13 shows a Web server in the DMZ connected to an existing WebSphere Application Server 5.0 in a secured zone. We are going to install Microsoft IIS plug-ins on a remote Web server and use an existing WebSphere Application Server 5.0 on `sandbox1.rigorconsultants.com` to install the portal. While there is an option to upgrade the WebSphere Application Server to a new version, the upgrade will not do the migration of security objects, deprecated API, cluster configuration, etc. It will just upgrade the server code.

1. On the Web Server, `http.rigorconsultants.com`, first check to see if the plug-ins have been installed in the `httpd.conf` file. If you see the following lines then go to step 12:

```
LoadModule ibm_app_server_http_module
"C:\WebSphere\AppServer/bin/mod_ibm_app_server_http.dll"
WebSpherePluginConfig "C:\WebSphere\AppServer/config/cells/plugin-
cfg.xml"
```

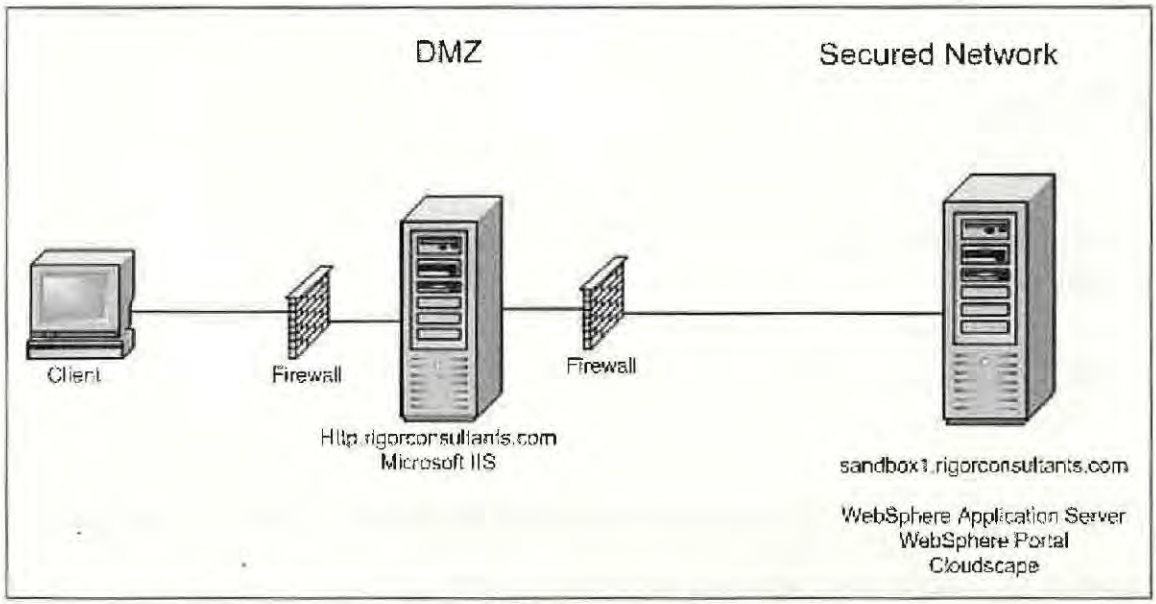

Figure 2-13 Installations on existing Web server and existing Web application server.

2. Stop the Web Server
3. Place the WebSphere Portal CD, which contains the WebSphere Application Server installation program (for Windows it should be CD 1.1 of WebSphere Portal V5.0), in http.rigorconsultants.com CD drive.
4. Launch the WebSphere Application Server installation entering in a command prompt drive:`\was\win\WAS50\install.exe`, where the drive is your CD drive. If you are installing WebSphere Portal 5.0, then WebSphere Application Server Enterprise Fixpack 1 (base plus frame) must be installed with the additional fixes required by WP. If you are installing WP 5.0.2, then WebSphere Application Server Enterprise Fix Pack 2 (base plus frame) must be installed with the additional fixes required by WP.
5. Select the language you want to use for the installation, and click OK.
6. Click Next on the welcome panel, indicate that you accept the license agreement, and click Next.
7. Select Custom.
8. On the features panel in Figure 2-14, deselect all features except Web Server Plugins and the entry for the specific Web server for which you are installing the plug-in. Click Next.

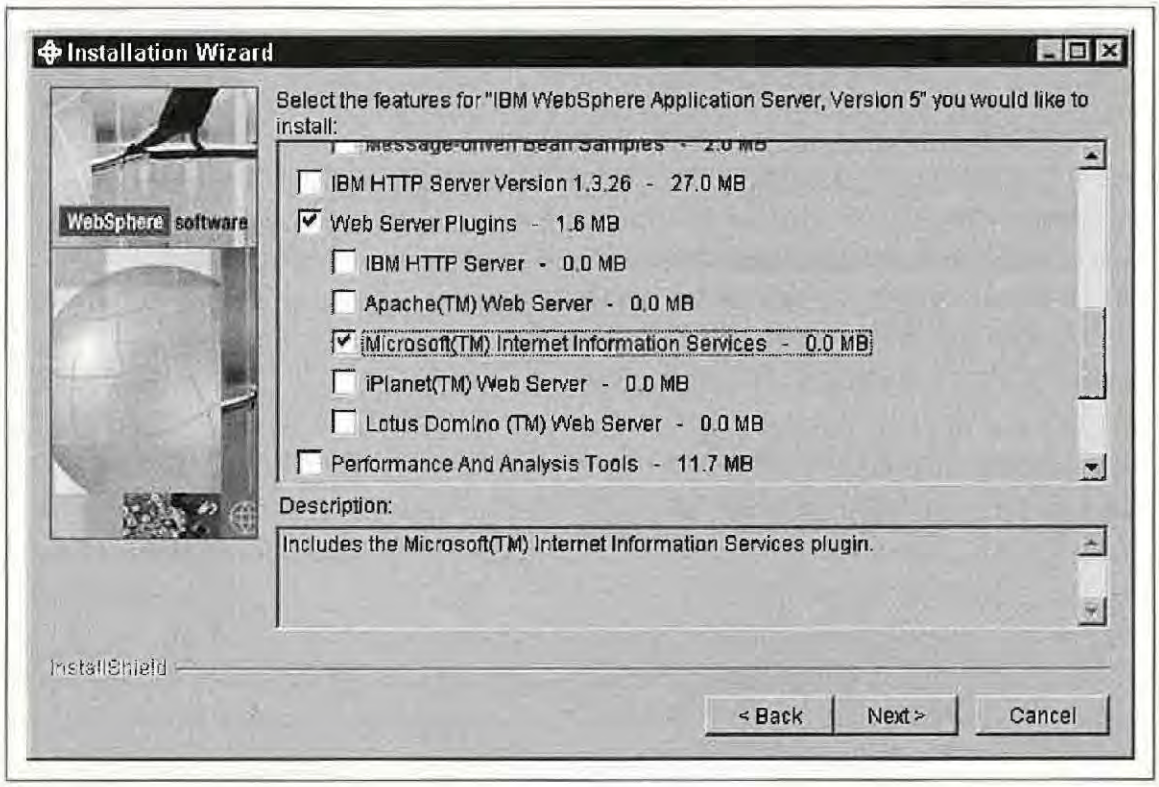

Figure 2-14 IBM WebSphere Application Server features panel.

9. Specify the directory path where you want to install the WebSphere Application Server files. This path must be the same as that used for the WebSphere Application Server installed on the local machine. Click Next.
10. Specify the location of the Web server configuration file, if requested. For IIS, it is not requested. Click Next.
11. Finish the installation.
12. Place CD in `sandbox1.rigorconsultants.com`.
13. Perform steps 1–7 in the previous section. Choose custom installation.
14. Select Configure to an Existing Install of WebSphere Application (see Figure 2-15). Click Next.
15. The installation program will then find the location of your WAS instance (as indicated in Figure 2-16). If it does not find it, click on Browse and locate the directory. Click OK and then Next.
16. The rest of the install is same as steps 11–14 of the quick install.
17. Verify the Web Server is running and create a back-up copy of `<wp_root>/config/wpconfig.properties file`.

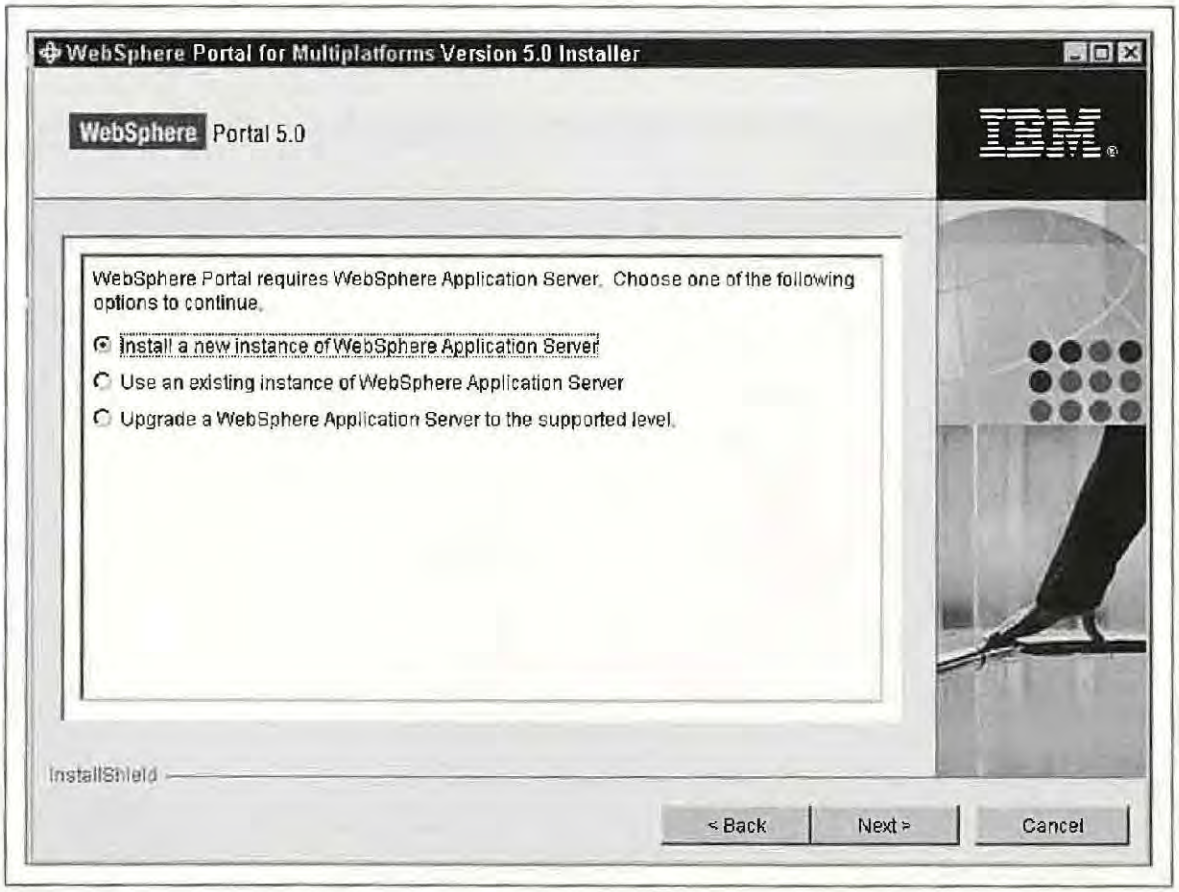

Figure 2-15 Choosing type of WAS installation.

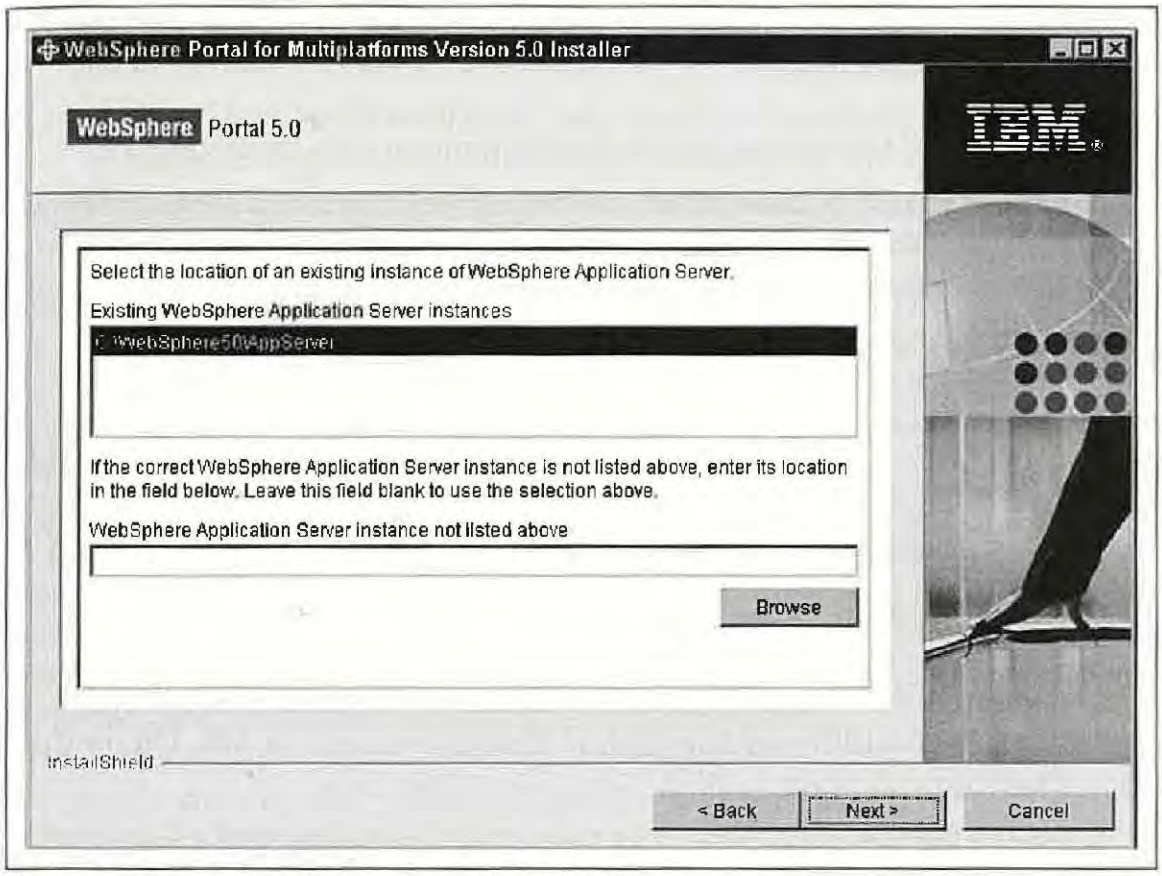

Figure 2-16 Finding the location of the WAS instance.

18. Using a text editor, open `<wp_root>/config/wpconfig.properties`, uncomment and set

    ```
    WpsHostName=localhost
    WpsHostPort=80
    ```

    WpsHostName is the fully qualified host name of the machine running WebSphere Portal while WpsHostPort is the port number that your Web Server is listening for HTTP traffic. You may not have to change any settings here.

19. Save the file, open a command prompt, and change the directory to `<wp_root>/config`.
20. Enter the command `WPSConfig.bat httpserver-config`.
21. Restart the Web Server on http.rigorconsultants.com and then restart the WebSphere Portal.

Installing WebSphere Portal Fixpack 2

In order to run WebSphere Portal 5.0.2, you have to install WebSphere Portal Fixpack 2. Assume that security is not installed and that this is a Windows

installation. Consult the readme files before you update. To do so, perform the following steps:

1. Stop your Web server and all WebSphere servers, including WP.
2. On the IBM site, download the WebSphere Application Server base fixpack 2 for Windows. It should be about 221 MB. Unzip the files and install the fixpack as per the instructions in the readme.
3. On the IBM site, download the WebSphere Application Server Enterprise Fix Pack 2 for Windows. It should be about 105 MB. Unzip the files and install the fixpack as per the instructions in the readme.
4. Restart the servers and ensure they successfully start. Then stop the servers.
5. On the IBM site, download the WebSphere Portal Fix Pack 2 for Windows. You need to download the fixpack 2 file. It should be about 359 MB. On the same download page, also download the WAS 5.0.2 additional fixes file. It should be about 14 MB. Unzip the files into a directory called `<wp_root>\update`.
6. Install the `WP_PTF_502.jar` using the UpdateWizard. Choose all the fixes and apply them. This applies to additional fixes to WebSphere Application Server so you can install WebSphere Portal 5.0.2.
7. Set up the Java environment by running `<was_root>\bin\setupCmdLine.bat`.
8. Open a command prompt and change the directory to `<wproot>\update`.
9. Enter the command, `updatePortal -fixpack -installDir "<wp_root>" -fixpackDir "<wp_root>\update" -install -fixpackID WP_PTF_502`
10. Upon successful completion, change the directory to `<wp_root>\config`.
11. Enter the command **WPSconfig.bat WP-PTF-502-DPortalAdminPWD=portalAdminPassword,** where portalAdminPassword is your WebSphere Portal administrator password. You must perform this step right after you have applied the fixes or the portal will be corrupted. This command must complete successfully. It will take quite a while to execute. If it does not complete successfully, then run `WPSconfig.bat UNINSTALL-WP-PTF-502 DPortalAdminPWD=portalAdminPassword,`

where `portalAdminPassword` is your WebSphere Portal administrator password, correct the problem, and redo this step.

12. Upon successful installation, start the Web Server and WebSphere Portal and log in. Check the version number on the WebSphere Portal Welcome Screen. It should now read IBM WebSphere Portal 5.0.2 in the lower left-hand corner.
13. Log in as an administrator. Click Document Manager and update the search index.

Summary

In this chapter, you have experienced the radically simplified WebSphere Portal installation, which provides a dramatic improvement on the customer experience. But most customer configurations require special installations configurations. The next chapter will explore installations for Linux platforms and examine more complex configurations.

CHAPTER

3

Installing WebSphere Portal on Linux

In the last chapter we showed how to install WebSphere Portal on a Windows 2000 Server platform. From the WebSphere Portal installation manual, much of the installation process can be extrapolated to the other platforms by making a few adjustments, right? It would be great if this were true, but it is not! The differences are a result of the lack of integration into the Linux user interfaces. The integration is more difficult due to numerous varieties of Linux user interfaces.

In this chapter you are going to learn how to install and uninstall WebSphere Portal on a Linux platform in either a graphics or a text environment. Linux was chosen due to the popularity of the platform and the close similarities to the zLinux, Solaris, and AIX installations.

Installing on Linux

So, you're going to install WebSphere Portal on Linux. First, let's deal with the urban legends. Despite what it says in the installation manual, WebSphere Portal does not work with all versions and permutations of Linux (see Figure 3-1). WebSphere Portal 5.0.2 has been certified with RedHat Enterprise Linux AS for Intel, RedHat Linux for Intel (x86) 8.0 2.4, RedHat Linux for zSeries 7.2 2.4, SuSE Linux for Intel (x86) 7.3 2.4, SuSE SLES for Intel (x86) 7 2.4, SuSE SLES for Intel (x86) 8 2.4, SuSE SLES for S/390 8 2.4, and SuSE SLES for S/390 7 2.4. For installation, you might be considering doing a full install. However, this will install IBM HTTP 1.3.26.1, which is not recommended for Unix systems. IBM HTTP 2.0.42.21 is based

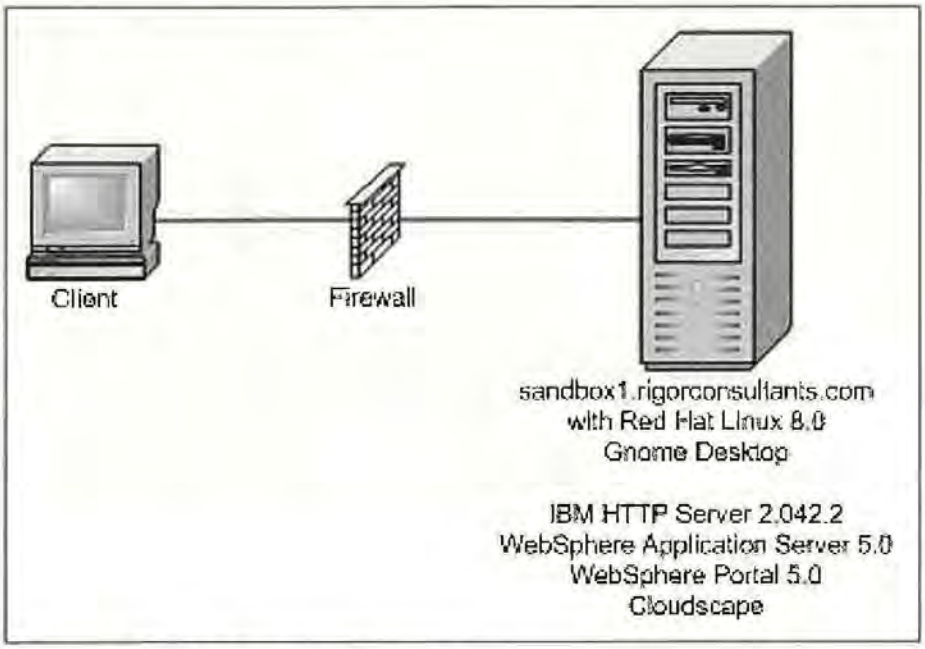

Figure 3-1 The Linux configuration.

on Apache 2.0, which enables for Unix threading in hybrid multiprocess, multithreaded mode. Realistically, most installations install the IBM HTTP Server on a separate server.

The installation you should be examining, which we will use as an example, is a custom installation using IBM HTTP 2.0.42.2 as the Web server and an installation of WAS 5.0 with WebSphere Portal 5.0 using Cloudscape on Red Hat 8.0 with the Gnome desktop.

The hardware requirements are as follows:

- A Pentium 4 processor at 1.4 GHz or higher
- 1 GB of RAM or more per processor (this is very important)
- 1.5 GB or more disk space on the root directory, 2.5 GB or more on `/opt` directory, and 500 MB or more for `/home` directory
- Network connectivity with a static IP address and a configured, fully qualified host name

Getting Information on Your Linux Server

Before you start, you need to perform some tasks on your Linux server to make it ready for the install. Specifically, perform the following tasks:

1. Identify your version of Linux.
2. Make sure you have a static IP address by going to Gnome Menu ⇨ System Settings and clicking Network Configurations. Highlight

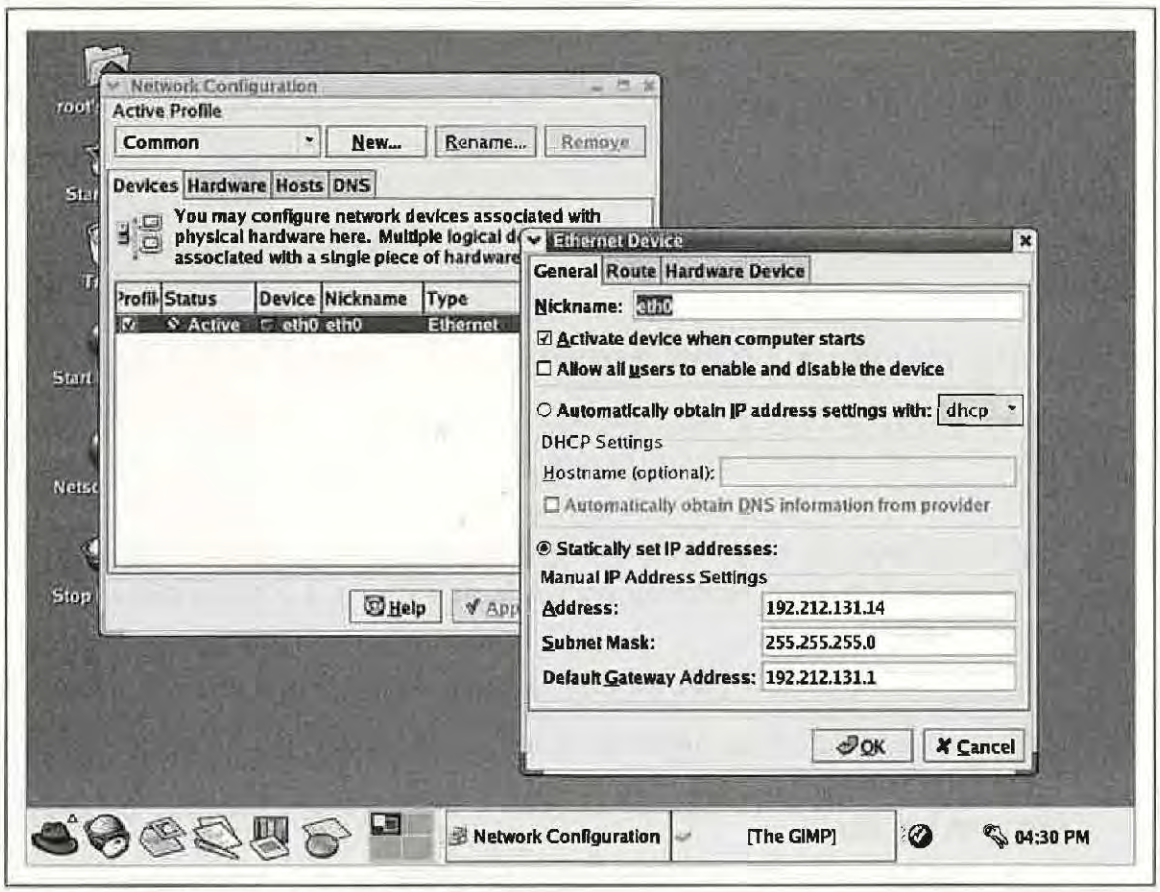

Figure 3-2 Network configuration utility.

Device and click Edit. Verify this is a static IP, as shown in Figure 3-2.

3. Right-click on the desktop and click New Terminal, and ping your server's fully qualified domain name and see if your IP comes up and four successful replies are returned.
4. Make sure that the following ports are not being used by typing **netstat-an** in a command prompt:
   - 443 (standard HTTPS port)
   - 523 (DB2 Administration Server)
   - 8008 (IBM HTTP Server Administration port)
   - 50000 (DB2 instance connection port)
   - 50001 (DB2 instance interrupt port)
   - 50002 (DB2 Control Server)
   - 55555 (WAS database port)
   - 9081 (Default Portal Port).
5. Make sure that user account has root or root authority.

6. Make sure no other Web server, such as Apache, is running. Remember, Red Hat installs Apache 1.3 even if you choose not to install the Web Server package.
7. Disable the Linux firewall.

You are now ready to install WebSphere Portal on Red Hat Linux 8.0.

Installing on Your Linux Server

To install WebSphere Portal, you first need to install a more current version of HTTP server than the one installed during a typical install:

1. Log in as root, go to the IBM Web site, and download the IBM Developer Kit, Java Technology Edition Version 1.3 and IBM HTTP Server 2.042.2.
2. Uncompress the IBM Developer kit and install it in `/opt`.
3. Edit your shell profile, append `/opt/IBMJava2-131/bin` to your PATH variable, export it, and reinitialize your environment (or log out and log in).
4. Uncompress the IBM HTTP install image in `/opt/ibmhttp`.
5. Open a new terminal, change directory to `/opt/ibmhttp`, and enter **java–jar setup.jar.**
6. Choose the language in which to run the installation. The Welcome screen appears (see Figure 3-3). Click Next.
7. The license agreement appears. Choose to accept and click Next.
8. Specify the directory name (see Figure 3-4). Choose the default `/opt/IBMIHS`. Click Next.
9. Select the type of installation (see Figure 3-5). Choose Typical, which installs everything. Click Next.
10. A window will appear summarizing your choices (see Figure 3-6). Click Next.
11. The installation then starts. A progress bar will appear with a message that states `Installing IBM HTTP Server`. Wait. If the progress bar has reached 100 percent and after five minutes the message has not changed to `Updating the RPM database`, the process has stalled. Hit Cancel, Reboot, Delete /opt/IBMIHS, and go back to step 5.
12. After the RPM database has been updated, click Finish.

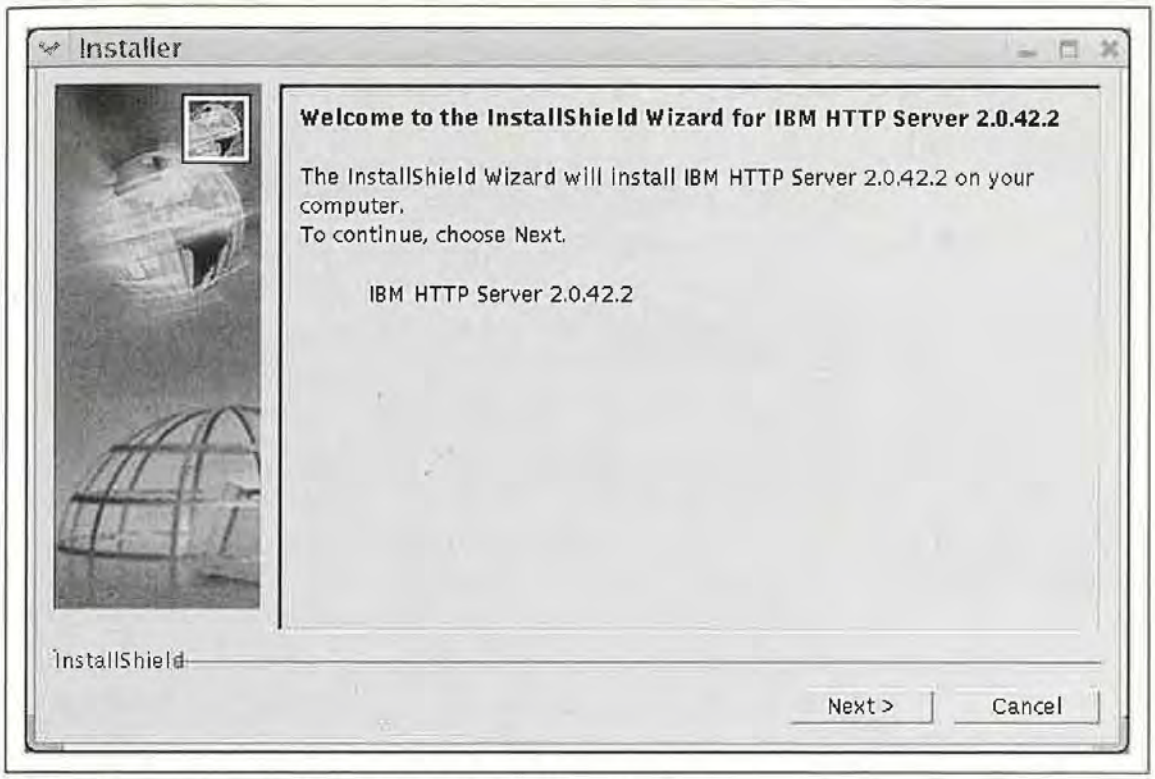

Figure 3-3 IBM HTTP Server 2.0.42.2 Welcome screen.

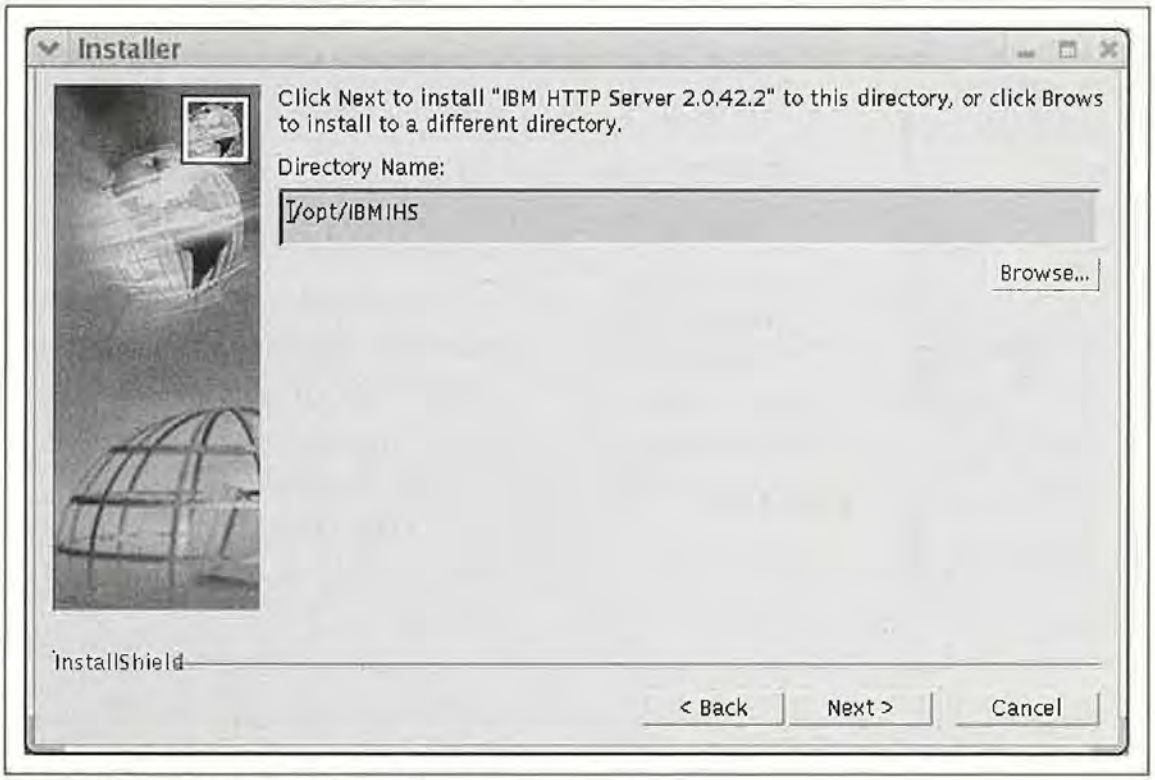

Figure 3-4 IBM HTTP Server directory.

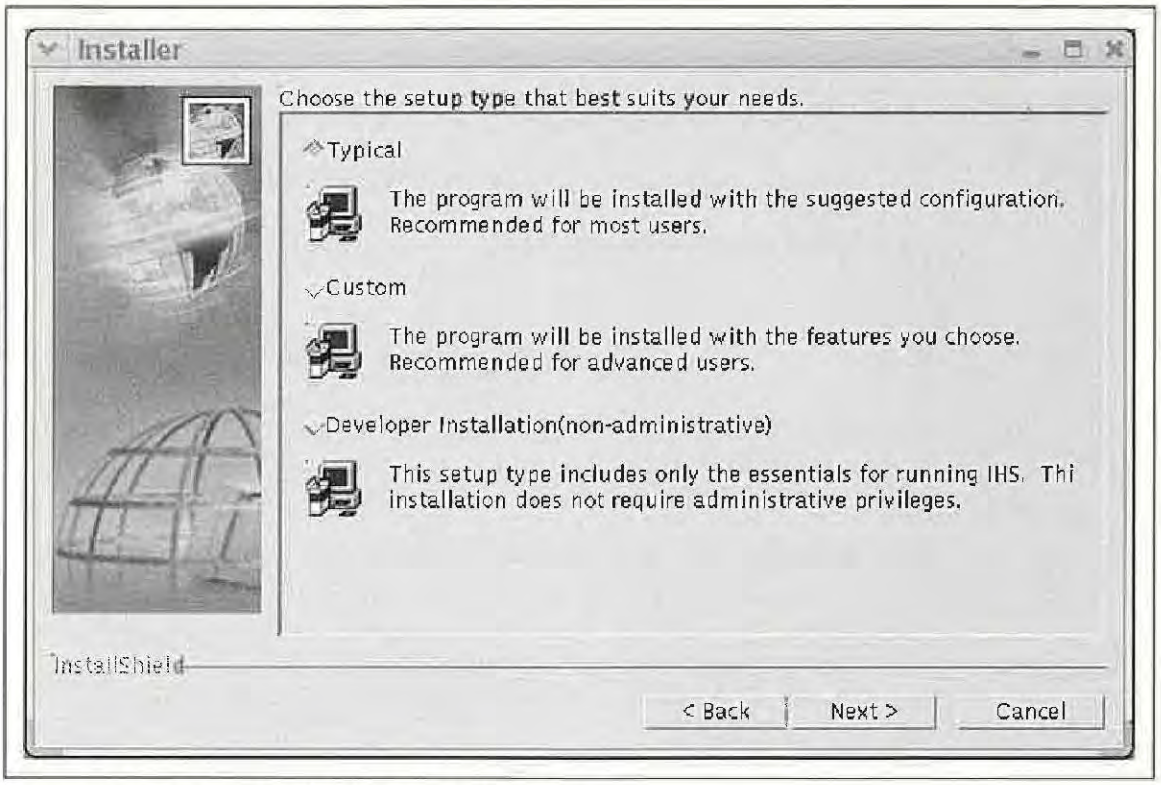

Figure 3-5 Choosing the type of installation.

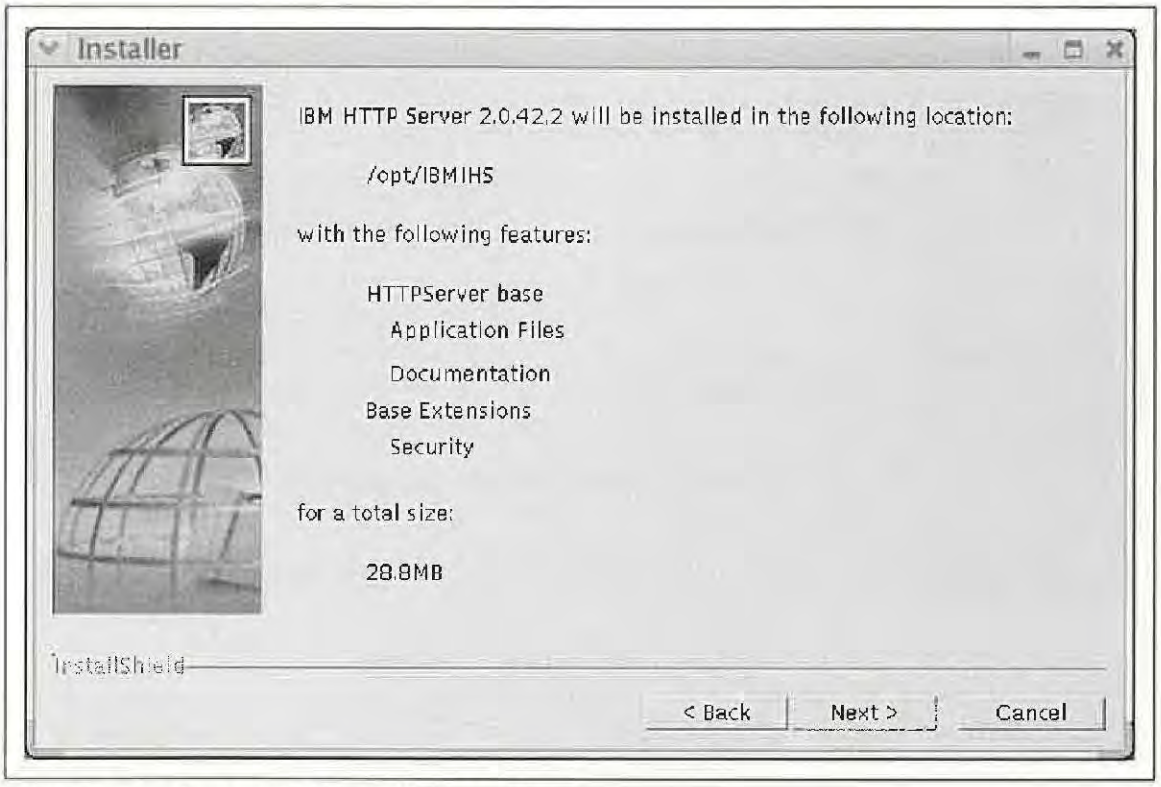

Figure 3-6 Summary of installation options.

Figure 3-7 IBM HTTP Server 2.0 Welcome screen.

13. Start the HTTP server by going into `/opt/IBMIHS/bin` and entering **./apachectl start**. Verify that the Web server is working by starting your Web browser and entering your fully qualified host name in the address bar (for example, `sandbox1.rigorconsultants.com`). You should see a page with **Welcome to IBM HTTP server** (see Figure 3-7).
14. Place the WebSphere Portal setup disc in the CD drive. Start a new terminal session and change the directory to `/mnt/cdrom`. Enter the command **./install.sh**.
15. Select the language for the install and click OK. Also close the terminal session so you will be able to eject the CD when the installation process requests you.
16. The Welcome panel will then be displayed. You will then be given the option to browse the infocenter if you choose Launch Infocenter. Otherwise click Next.
17. The license agreement will appear. Select `I accept the terms in the license agreement` and click Next.
18. The installation then checks the machine for operating system and software prerequisites. Click Next.

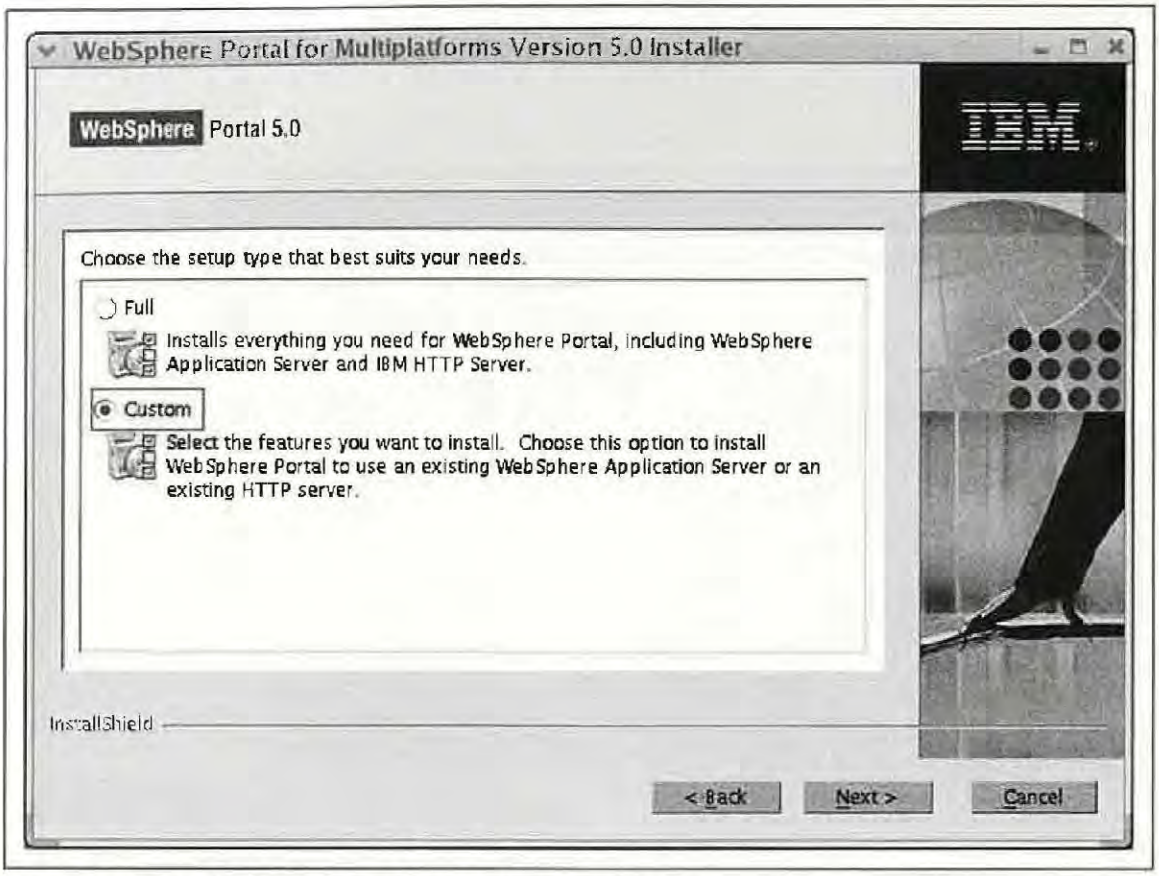

Figure 3-8 WebSphere Portal installation type screen.

19. You will then be asked if you want a Full or Custom install (see Figure 3-8). Choose Custom.
20. You will then be asked if you want to install a new WebSphere Application or configure to an existing install of WebSphere Application. Click on New Install and select Next.
21. You will then be asked to specify the directory to install WebSphere Application Server (see Figure 3-9). If the directory does not exist, it will be created. By default, IBM suggests that you install in `/opt/WebSphere/AppServer`. Enter the directory and click Next.
22. The next screen asks you if you want to install IBM HTTP Server or configure WAS to use an existing Web server (see Figure 3-10). Choose `Install the plug-in for an Existing Http Server`. Click Next.
23. You will then be asked to choose the type of HTTP server to use with WAS (see Figure 3-11). Click IBM HTTP Server. Click Next.
24. The next screen will ask you to specify the location of the http configuration file (see Figure 3-12). Enter **/opt/IBMIHS/conf/httpd .conf** and click Next.

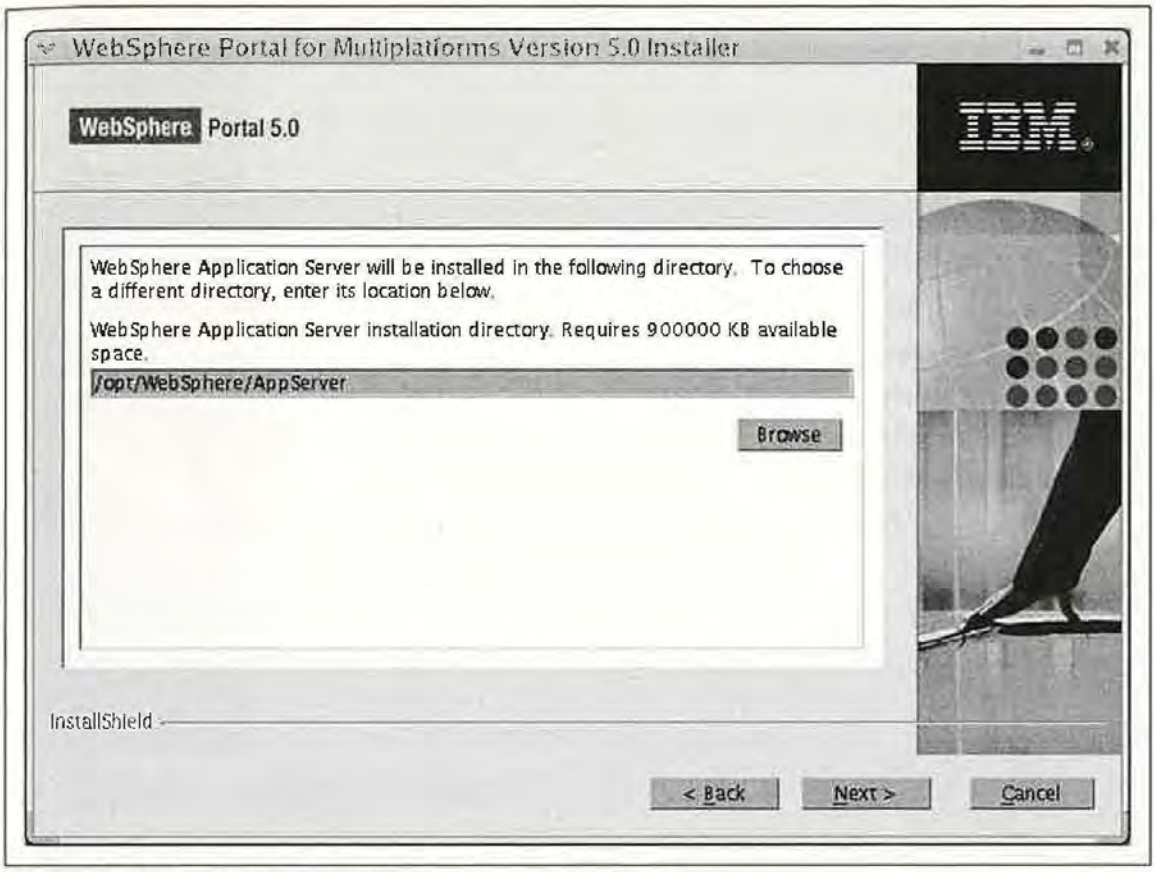

Figure 3-9 WebSphere installation directory screen.

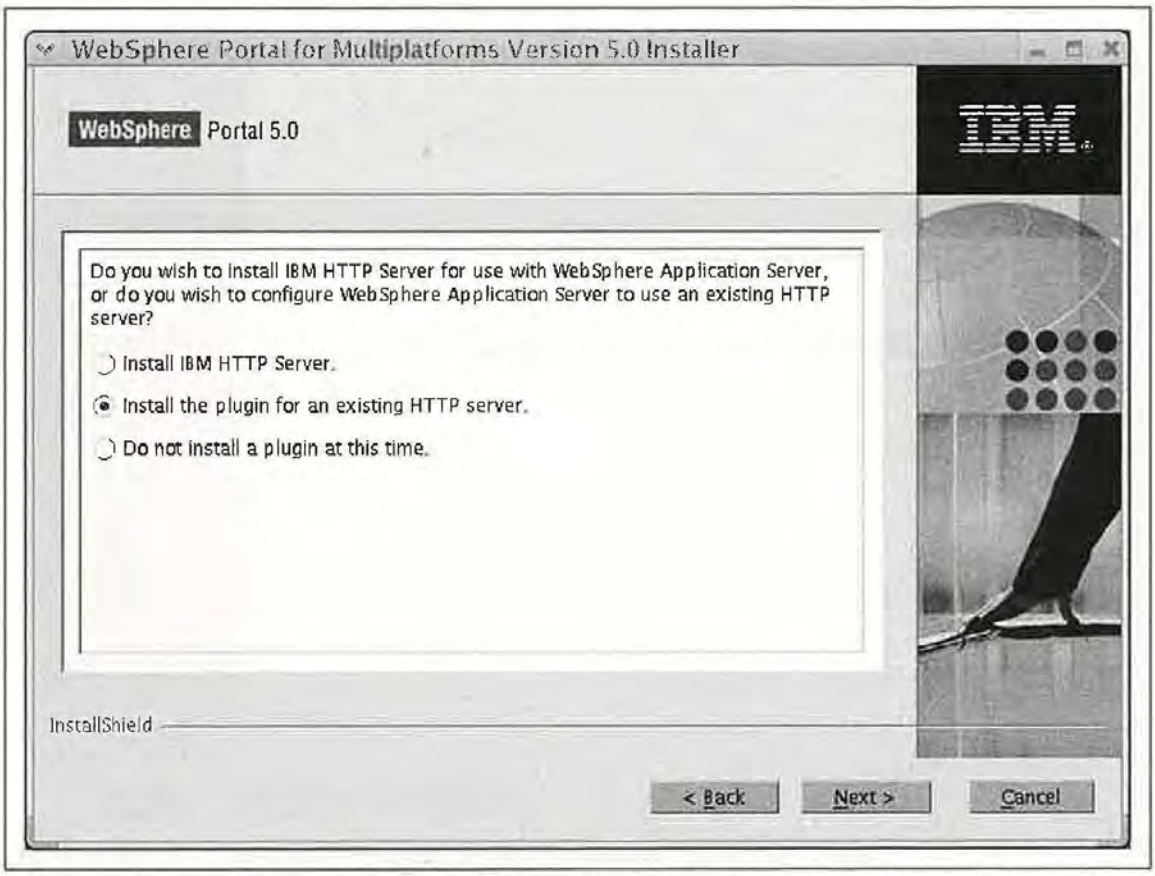

Figure 3-10 Installing a new HTTP server or configuring an existing Web server.

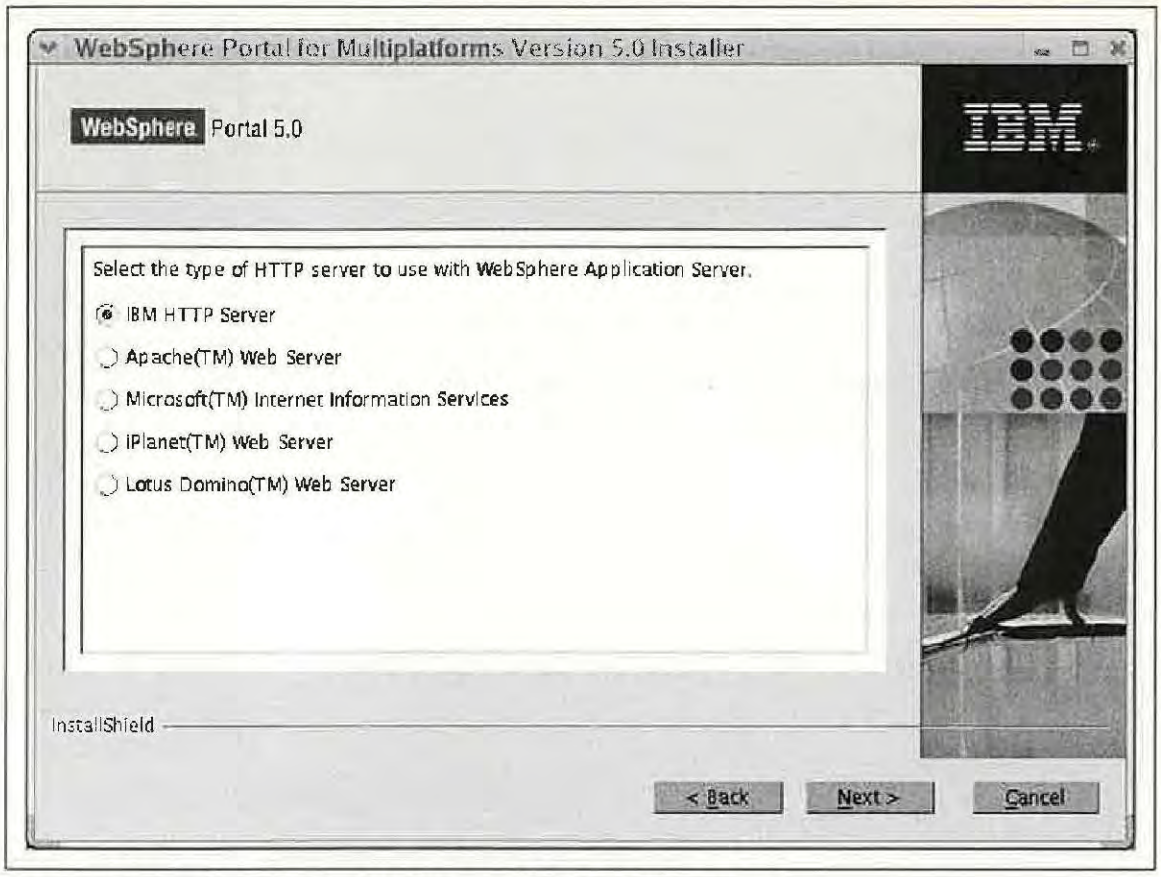

Figure 3-11 Choosing the HTTP server to install the plug-in.

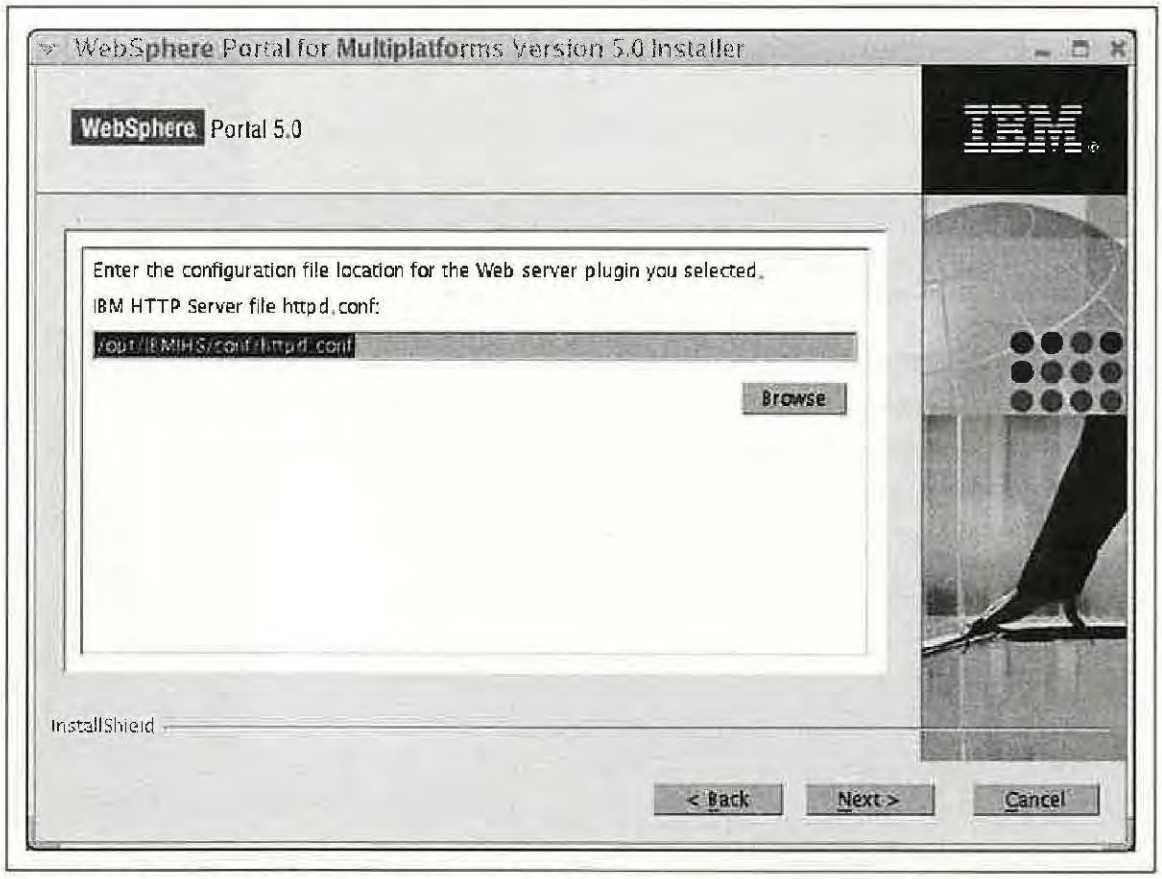

Figure 3-12 Specifying the location of the HTTP server configuration file.

25. You will then be asked where you want to install WebSphere Portal. Usually it is installed off the WebSphere Application Server directory root or `/opt/WebSphere/Portalserver`. Click Next.
26. Enter the user ID and password for the WebSphere Portal Administrator. (see Figure 2-9). The password and user ID must be at least five letters long and contain no blanks. This user ID is specific to the WebSphere Portal and does not have to be an Operating System User ID but it must conform to the rules of the LDAP if you are using an LDAP to manage your users.
27. Verify the components to be installed and click Next.

Next, the program starts installing the selected components. You see the progress indicator moving as the components are installed. When it asks you for a CD, you will have to right-click on the CD icon and click Eject. When the installation is finished, a confirmation panel, such as the one shown in Figure 3-13, is displayed with a listing of the components installed. The confirmation panel will also display the port number used to access the portal. The port number is also stored in the WpsHostport

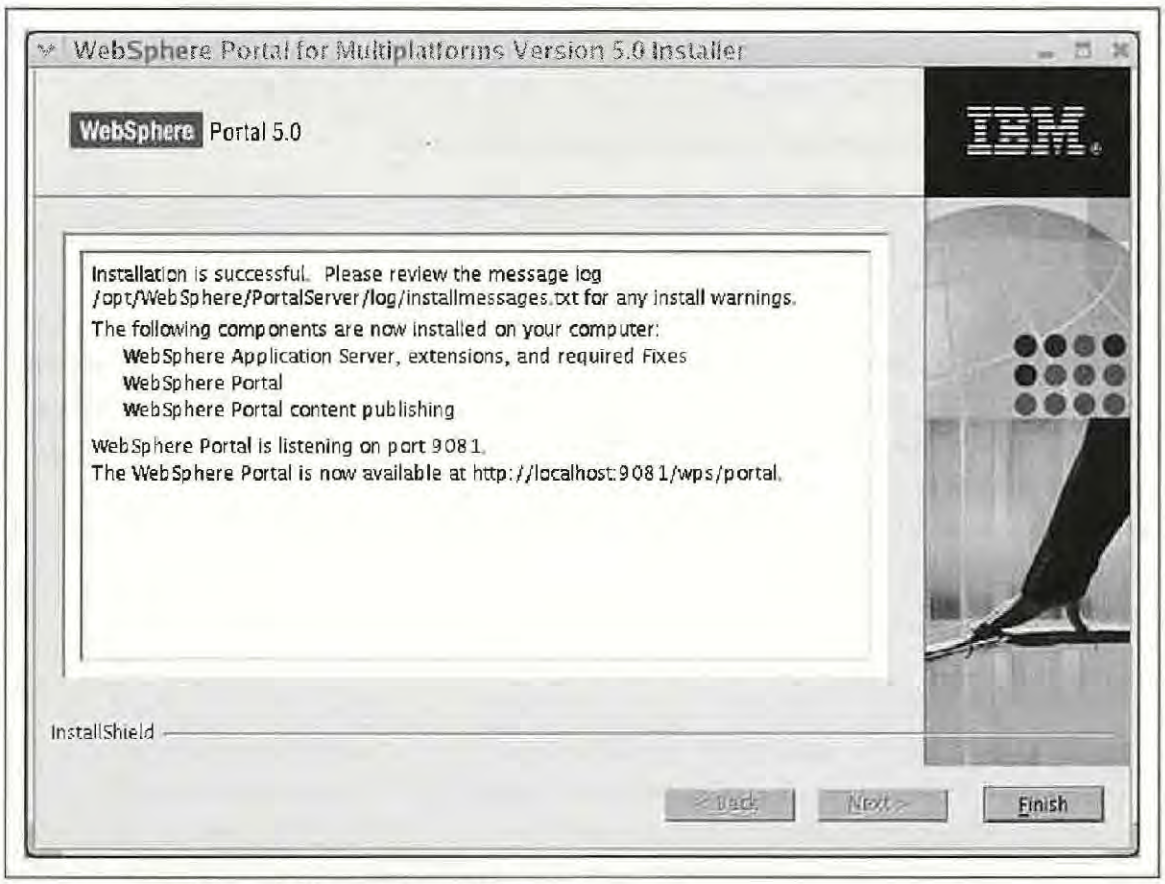

Figure 3-13 Screen after a successful install.

property in <wp_root>/config/wpconfig.properties file. Record the port number and click Finish.

In Linux, the installation program does not add menu items to the desktop. You will need to do this manually. You will also need to apply fixpack 2. Use the instructions in Chapter 2 except for downloading the Linux fixpacks and use forward slashes instead of backward slashes for the commands. Remember to replace .bat with .sh for the script commands.

The WebSphere Portal installation modifies the Web server configuration file to support IBM HTTP 1.3.26.1 Web server plug-ins. To correct the entries to support IBM HTTP 2.042.2, go to the http.conf file in /opt/IBMIHS/conf and change:

```
LoadModule ibm_app_server_http_module
/opt/WebSphere/AppServer/bin/mod_ibm_app_server
```

to

```
LoadModule was_ap20_module
/opt/WebSphere/AppServer/bin/mod_was_ap20_http.so
```

Next you have to configure WP to use the IBM HTTP Web Server instead of the WebSphere Internal HTTP server, which is only to be used for development or to show prototypes. You must do the following steps to configure WP to use IBM HTTP Server:

1. Open a terminal session and go to <wp_root>/config.
2. Edit wps.properties and set WpsHostPort=80.
3. Enter **./WPSconfig.sh httpserver-config**.
4. Wait for the Build Successful message.
5. Restart the Web server and WebSphere Portal.

Next create a service for IBM HTTP Server so that it will start at startup. In /etc/rc.d/init.d, create a script file called ibmhttp with permissions 755 and insert the following code. The code will support start, stop, restart, and provide status for the IBM HTTP Web Server.

```
#!/bin/bash
#
# Startup script for the IBM Http Web Server
#
# chkconfig: - 86 16
# description: IBM HTTP is a World Wide Web server. It is used to
serve \
#            HTML files and CGI.
# processname: ibmhttpd
# config: /opt/IBMIHS/conf/httpd.conf
```

```
# Source function library.
. /etc/rc.d/init.d/functions

apachectl=/opt/IBMIHS/bin/apachectl
httpd=/opt/IBMIHS/bin/httpd
RETVAL=0

case "$1" in
start)
echo "Starting IBM HTTP Web Server..."
$apachectl -k start
RETVAL=$?
    ;;
stop)
echo "Stopping IBM HTTP Web Server..."
$apachectl -k stop
RETVAL=$?
    ;;
restart)
echo "Restarting IBM HTTP Web Server..."
$apachectl -k restart
RETVAL=$?
    ;;
status)
status $httpd
    ;;
  *)
echo "Usage: $0 (start |stop|restart|status)"
    ;;
esac
exit $RETVAL
```

Next run `chkconfig--add ibmhttpd` in a terminal session to create links to all the appropriate run levels. Then go to Server Settings ➪ Services, refresh the list, check ibmhttp, click Start, and click Save.

Do the following to verify the installation was correct:

1. Make sure the IBM HTTP Server is running.
2. Start the portal server by going to a terminal session, changing the directory to `/opt/WebSphere/AppServer/bin` and entering **./startServer.sh WebSphere_Portal**.
3. Enter in the address bar in your browser the URL `http://<hostname.yourco.com>:,port>/wps/portal`, where hostname.yourco.com is the WebSphere Portal fully qualified host name and port is the port number in the confirmation panel (usually 9081). Click Go and you should see the WebSphere Portal Welcome page similar to Figure 2-12.

Nongraphic Linux Installation

Very often in Linux environments, installations are done on servers through telnet or the console. WebSphere Portal provides support for these environments using a text-based interactive installation.

To perform a text-based WebSphere Portal installation, copy the Setup CD-ROM to a location on your hard drive. Change into the directory and enter **./install.sh –console** (**\install.bat –console** for Windows). An output similar to Figure 3-14 should appear.

You can also use a response file. A response file contains the values that you use during an install and is also used to create a standard WebSphere Portal installation for multiple machines. The values can also be passed as parameters. To run an installation program with a response file enter `./install.sh--options response_file` where response-file is the name of the response file. For Windows it is `\install.bat--iptions response_file`.

A response file can be manually created or automatically during an install. The information entered during an installation is automatically saved in `<was_root>/PortalServer/Log/responselog.txt`. The

```
root@sandbox1:/cdrom
File  Edit  View  Terminal  Go  Help
.................................
........................
-------------------------------------------------------------------------------
Select a language to be used for this wizard.

[ ] 1  - Czech
[X] 2  - English
[ ] 3  - French
[ ] 4  - German
[ ] 5  - Greek
[ ] 6  - Hungarian
[ ] 7  - Italian
[ ] 8  - Japanese
[ ] 9  - Korean
[ ] 10 - Polish
[ ] 11 - Portuguese
[ ] 12 - Portuguese (Brazil)
[ ] 13 - Russian
[ ] 14 - Simplified Chinese
[ ] 15 - Spanish
[ ] 16 - Traditional Chinese
[ ] 17 - Turkish

To select an item enter its number, or 0 when you are finished: [0]
```

Figure 3-14 Console installation screen.

following is an edited version of the response log created by the Linux Install. Even though this abridged file is quite lengthy, we are showing it because the comments describe the syntax and semantics of the installation commands that are used within a response file. For space reasons, many of the comments have been edited from this example.

```
# Response File for WebSphere Portal Version 5.0 Silent Installation
#
# IMPORTANT: ALL VALUES MUST BE ENCLOSED IN DOUBLE QUOTES ("").
#
#
# INSTALL WEBSPHERE APPLICATION SERVER
#
# Description: Indicate whether you want to install a new instance of
WebSphere
# Application Server or use an existing instance.
# Values:
#
#   install - Install a new instance of WebSphere Application Server.
#   use     - Use an existing instance of WebSphere Application Server.
#   migrate - Upgrade an existing WebSphere Application Server to the
supported
#             level.

     -W installWas.choice="install"

#
# WEBSPHERE APPLICATION SERVER INSTALLATION LOCATION
#
# Description: Specify the directory where you want to install WebSphere
# Application Server.
     -W was.location="/opt/WebSphere/AppServer"

# INSTALL WEB SERVER (IBM HTTP SERVER)
#
# Description: Indicate whether you want to install a new Web server or
use an
# existing Web server.
#
# Values:
#   install - Install IBM HTTP Server
#   use     - Use an existing Web server

     -W installIhs.choice="use"

# IBM HTTP SERVER INSTALLATION LOCATION
#
# Description: Specify the directory where you want to install IBM HTTP
Server.
# Application Server.
```

```
     -W ihs.location="/opt/IBMHttpServer"

# HTTP SERVER TYPE
#
# Description: Specify the type of Web server that you want to use with
# WebSphere Portal.
#
# Values:
#   ihs     - IBM HTTP Server
#   apache  - Apache(TM) Web Server
#   iis     - Microsoft(TM) Internet Information Services
#   iplanet - iPlanet(TM) Web Server
#   domino  - Lotus Domino(TM) Web Server


     -W httpServerType.choice="ihs"

#==============================================#
# IBM HTTP Server Configuration File Location #
#==============================================#

     -W ihsPlugin.file="/opt/IBMIHS/conf/httpd.conf"

# WEBSPHERE APPLICATION SERVER NODE NAME
#
# Description: Specify the node within the WebSphere Application Server
cell to
# which the WebSphere Portal application server will belong.
     -W node.name="sandbox1"

#
# WEBSPHERE APPLICATION SERVER HOST NAME
#
# Description: Specify the fully qualified host name or IP address of
# the computer running WebSphere Application Server. For example,
# "hostname.yourco.com".
#

     -W node.hostName="sandbox1.rigorconsultants.com"

#
# Begin Installing Services
#


# WEBSPHERE PORTAL INSTALLATION LOCATION
#
# Description: Specify the directory where you want to install WebSphere
Portal.
#
```

```
     -W portal.location="/opt/WebSphere/PortalServer"

# WEBSPHERE PORTAL ADMINISTRATIVE USER AND PASSWORD
#
# Enter the user ID and password for the Portal
# administrator

     -W portalAdmin.user="wpsadmin"
     -W portalAdmin.password="PASSWORD-REMOVED"
# SETUP CD LOCATION
#
# Description: Specify the directory path to the Setup CD.

     -W cdSetup.cdPath="/mnt/cdrom"

#
# WEBSPHERE APPLICATION SERVER CD LOCATION
#
# Description: Specify the directory path to the WebSphere Application
Server
# installation images.
     -W userInputCDLoc2.cdPath="/mnt/cdrom"

# WEBSPHERE APPLICATION SERVER FIXPACK AND EFIXES CD LOCATION
#
# Description: Specify the directory path to the WebSphere Application
Server
# Fixpack and eFixes installation images.
     W wasfix1MediaLocation.cdPath="/mnt/cdrom"

#
# WEBSPHERE PORTAL CD LOCATION
#

     -W WPSCDLoc.cdPath="/mnt/cdrom"


#
# PORTAL BASIC CONFIGURATION OPTION
#
# Description: Specify whether you want to perform basic
configuration of
# WebSphere Portal automatically when WebSphere Portal is installed.
# Values:
#
#   yes - Basic configuration is performed automatically as part of the
#         WebSphere Portal installation. This is the default value and
is
#         assumed if the basicConfig.choice parameter is not specified.
#
```

```
#   no  - Basic configuration is not performed during installation and
must be
#         performed manually after installation.

# -W basicConfig.choice="no"

# The options above make this a custom install

    -W setupTypePanel.selectedSetupTypeId="custom"
```

If you have problems creating the response file then before you install, copy the setup CD to a local drive and CHMOD 777, the `installresponse.txt` file. The file `installresponse.txt` is a template of the response file.

Uninstalling WebSphere Portal on Linux

To uninstall WebSphere Portal on Linux, go to `<was_root>/PortalServer/uninstall` and execute `./uninstall.sh`. To uninstall the IBM HTTP Server, go to `<ihs_root>/_uninst` and enter **java -jar uninstall.jar.**

NOTE Even though the IBM Http installed the IBM Global Security Kit, it will not uninstall it. To uninstall the Global Security Kit, find the name by entering `rpm -qa | grep gsk`; then `rpm -e the package`.

But what happens if the uninstall does not work, is corrupt, or is accidentally deleted? Then, a manual uninstall is necessary; however, this only works if you uninstall both WebSphere Portal and WebSphere Application Server. We are also assuming that you do not have any data that you want to save. To perform a manual uninstall perform the following steps:

1. Create a backup.
2. Kill all Java processes by typing the command **killall -9 java.**
3. Edit in the root home directory the `vpd.properties` file and remove any entries starting with WS.

Directory Structure

To understand the WebSphere Portal installation in more detail, let's see exactly what it creates. In Windows, installation places the information in the directories as an extension of `<was_root>\PortalServer` while in

Linux/Unix, they are an extension of <was_root>/PortalServer. The following describes the WebSphere Portal directory structure:

```
wp_root                    Root directory for WebSphere Portal
|
•     uninstall                  Files used to uninstall WebSphere Portal
|
+-- app Transcoding tag library definition in the /wps.ear/wps.war/WEB-
INF/tld subfolder
|
+-- bin                    WebSphere Portal tools
|
+-- cloudscape             Cloudscape database files
|
+-- config Portal          configuration files
|
+-- deployed               Copies of the .ear and .war files for
installed portlet applications.
|
+-- dev                    Resources for portlet developers, including
Struts
|
+-- doc WebSphere          Portal InfoCenter and Javadoc
|
+-- IBMTrans               Transcoding component
|
+-- installableApps        WAR files prior to deployment
|
+-- installedApps          Active portlet applications extracted to the
WAR file directory structure
|
+-- launchPad              Images used by the WebSphere Portal First
Steps
|
+-- license                WebSphere Portal license agreement
|
+-- log                    WebSphere Portal log files
|
+-- migration              Scripts used to assist in migrating from
previous releases of WebSphere Portal
|
+-- odc                    On-Demand Client files
|
+-- package                Response files and utilities for install
|
+-- pb                     Property broker files for cooperative portlets
|                          XML files for installing portlets individually
+-- portletscripts
|
+-- shared
```

```
| |
| +-- app WebSphere Portal runtime JARs, TLDs, and other resources
| |
+-- config Portal configuration files
| | |
| |
+- services Portal services configuration files (*.properties)
| |
| +-- nls WebSphere Portal NLS files
|
| +-- WEB-INF Resources for the WebSphere Portal
| |
| +- tld Portal JSP taglibs
|
+-- uninstall Resources for uninstalling WebSphere Portal and components
|
+-- version Version information for various components
|
+-- wmm Member Manager configuration, including attributes of portal
users
|
+-- wpcp WebSphere Portal Content Publisher runtime and resources
|
+-- wts Resources for Transcoding Technology
```

The WebSphere Portal applications are stored in the WAR directory under `<was_root>/AppServer/installedApps/<hostname>/wps.ear/wps.war`:

```
wps.war
|
+-- c2a Cooperative portlet resources
|
+-- dtd Document Type Definitions (DTDs) for Portal Server
|
+-- html HTML files for the portal
|
+-- images Graphics for the portal
|
+-- menu Files for MenuService
|
+-- META-INF Metadata for the portal Web application
|
+-- peopleawareness Files for PeopleService
|
+-- screens Screen JSPs for the portal
| |
| +--  markup_name Subdirectory for each markup type
|
|-- skins Skin JSPs for the portal
```

```
| |
| +--  markup_name Subdirectory for each markup type
|-
|-- themes Theme JSPs for the portal
|
| +--  markup_name Subdirectory for each markup type
|
+-- WEB-INF Resources for the Portal Server Web application
```

Installing When WebSphere Security Is Enabled

In most production environments, WebSphere security will be enabled. Installing WebSphere Portal with WebSphere security can be unnecessarily challenging, so disabling security temporarily is highly recommended.

Using the WebSphere administrative console, disable security, install WebSphere Portal, configure WebSphere Portal to use WebSphere Application Server security (discussed in great detail in Chapter 19), and then re-enable WebSphere Application Server Security.

Summary

In this chapter you learned the tricks to installing WebSphere Portal in a Linux environment for both graphic and console desktops. However, all of our installations have dealt with "simplistic" installations. In the next few installation chapters, we are going to discuss how to configure WebSphere Portal with more robust databases and to integrate the various supporting products.

CHAPTER

4

Customizing WebSphere Portal

You have now installed WebSphere Portal on both your Windows and Linux platforms. You are amazed how much easier the installation was compared to WebSphere Portal 4.X. A few questions to answer, pop in four CDs, and—voilà!—WebSphere Portal is installed. Now you might be looking at the other 58 CDs and wondering, "what are these for?" Like a person who has put together a bicycle and finds a few parts left over, you wonder if you missed something.

Well you haven't! To simplify the installation process, WebSphere Portal mandates that the basic configuration is installed and then the user modifies it using the tool WPSconfig. You use WPSconfig when you want to do the following:

- Change the database from Cloudscape
- Use a Directory server
- Use a remote Web server

The reason for this design decision is to enable the user to modify the configuration without having to reinstall the program. This capability clearly makes sense. Initially, when you install the program, you are testing out the features, writing some small portlets, and maybe doing a couple of pilots. For these requirements, you really don't need security and Cloudscape is suitable as a database. However, when you want to roll out a real application with scalability requirements you will need to add a security model and a database that satisfies your load, change, and operation requirements. With this new installation model, you will just need to run WPSconfig to change the database and/or directory server configuration and migrate the data. You will not need to reinstall the product.

WPSconfig

WPSconfig is a script file that invokes configuration tasks. The configuration tasks are implemented using the Apache Ant Tool, which is included as part of WebSphere Application Server.

Apache Ant (`http://ant.apache.org/index.html`) is a Java-based build tool whose functionality is extended using Java classes instead of scripts, as for most other build tools. Ant's build files are written in XML. They define the project and the tasks that need to be performed during the configuration. Configuration values are passed as parameters during invocation or through property files.

Each build file contains one or more projects. The project tag defines the name of the project, the default target to use when no target is supplied, and optionally the base directory where all path calculations are done. Each project has one or more target(s). A target is a set of tasks that need to be executed. On the target you can specify the order in which a target gets executed or a property that must be set (or not) for it to be executed.

A project can have a set of properties that are either set in the build file by the property task or can be set externally to Ant.

WPSConfig is invoked by typing the following:

```
* Unix: <wp_root>/config/WPSconfig.sh task_name [task_name...]
{-Dproperty =value]
* Windows: <wp_root>\config/WPSconfig.bat task_name
[task_name...] {-Dproperty =value]
```

`task_name` is the configuration task and `-Dproperty` is the property flag that enables you to pass the property value to be used by the task. Normally you would set the property values in `WPSconfig.properties` rather than pass them as parameters. When a value is invoked from property parameter, it temporarily supersedes the value set in `WPSconfig.properties` but does not overwrite it. The precedence order (highest to lowest) for setting properties is as follows:

1. Properties set on the command line
2. Properties set in parent properties file
3. wpconfig.properties
4. WPSconfig.xml

WPSconfig script invokes the Ant build tool, which receives its project and configuration task definitions from `WPSconfig.xml`. WPSconfig defines the configuration tasks you can invoke; see Table 4-1 for a description of the tasks.

Table 4-1 WebSphere Portal Configuration Tasks

TASK NAME	DESCRIPTION
enable-security-ldap	Configures the WPS50 and WMM databases for use with an LDAP directory. Sets WebSphere Application Server security to use the LDAP directory
database-transfer-export	Exports existing data from Cloudscape
database-transfer-export-linux	Exports existing data from Cloudscape on Linux
create-local-database-db2	Creates all the databases needed by WebSphere Portal and WebSphere Portal content publishing for a locally installed DB2 instance
database-transfer-import	Imports data collected by the database-transfer-export task
httpserver-config	Configures WebSphere Portal for use with HTTP server. The WpsHostPort value must be set first in the `wpconfig.properties` file

Configuration Templates

WebSphere Portal derives the values for the configuration tasks from the `wpconfig.properties` file. This file contains a centralized reference for all the configuration settings. However, this file contains a large number of properties, many of which the user does not need to change. To make the property file more workable, WebSphere Portal introduces the concept of configuration templates, which are special-purpose property files that contain the properties for a specific task.

Configuration templates is a parent property file. A parent property file has values that when invoked supersede the same property value in the `wpconfig.properties` file. A parent property file can also be a copy of `wpsconfig.properties` with set values. This can be used to define a common configuration over multiple machines.

In the `<wp_root>/config/helpers` directory, WebSphere Portal provides configuration templates. See Table 4-2 for a description of these templates.

To use the configuration templates, copy the template from `<wp_root>\config\helpers` to `<wp_root>\config`. Edit the file and change the property values to those that are appropriate to your environment (later on we will expand this much further). In the template, above the value are comments that go into great detail on what the value should be and what the default value is. More often than not, the default value does not need to be changed.

Table 4-2 WebSphere Portal Configuration Templates

CONFIGURATION TEMPLATE	DESCRIPTION
`transfer_db2.properties`	Configures WebSphere Portal to use a DB2 database, including transferring your data from Cloudscape to DB2
`transfer_informix.properties`	Configures WebSphere Portal to use an Informix database, including transferring your data from Cloudscape to Informix
`transfer_oracle.properties`	Configures WebSphere Portal to use an Oracle database, including transferring your data from Cloudscape to Oracle
`transfer_sqlserver.properties`	Configures WebSphere Portal to use an SQL Server database, including transferring your data from Cloudscape to SQL Server
`config_http.properties`	Configures WebSphere Portal to use an external Web server, instead of the internal HTTP transport provided by WebSphere Application Server
`security_active_directory.properties`	Configures WebSphere Portal to use WebSphere Application Server security with Active Directory as its Lightweight Directory Access Protocol (LDAP) directory
`security_disable.properties`	Disables security for WebSphere Application Server if you have previously configured WebSphere Portal to use security
`security_domino.properties`	Configures WebSphere Portal to use WebSphere Application Server security with Domino as its LDAP directory
`security_ibm_dir_server.properties`	Configures WebSphere Portal to use WebSphere Application Server security with IBM Directory Server as its LDAP directory
`security_sun_one.properties`	Configures WebSphere Portal to use WebSphere Application Server security with Sun ONE as its LDAP directory

After you have modified the configuration template, execute WPSConfig passing the configuration template as a parent properties file using the following syntax:

```
* UNIX: ./WPSconfig.sh -
DparentProperties=<wp_root>/config/<configuration_template> -
DSaveParentProperties=true
*Windows: WPSconfig.bat -
DparentProperties=<wp_root>\config\<configuration_template> -
DSaveParentProperties=true
```

Note that when you call WPSconfig, you are not specifying a task. When you do not specify a task, the configuration program updates the `wpconfig.properties` file with values from the configuration template. Of course, this happens if and only if the configuration tasks execute successfully.

Of course, you do not have to use the configuration templates provided by IBM. You can create your own. Just copy a configuration template that matches as close as possible to the configuration you want to make and modify the file adhering to the conventions specified in the file.

Using DB2 with WebSphere Portal

Now that you understand how WPSconfig works, you need to know how to modify the WebSphere Portal for a real-life scenario. In Figure 4-1, in the DMZ, the WebSphere Portal is on Windows 2000 Advance Server on `sandbox1.rigorconsultants.com`. In the secured zone, data is stored on DB2 UDB 8.1 Enterprise database server. In this configuration, the data is stored in a more robust database with features to support enterprise operation management, backup and recovery, performance turning, and change

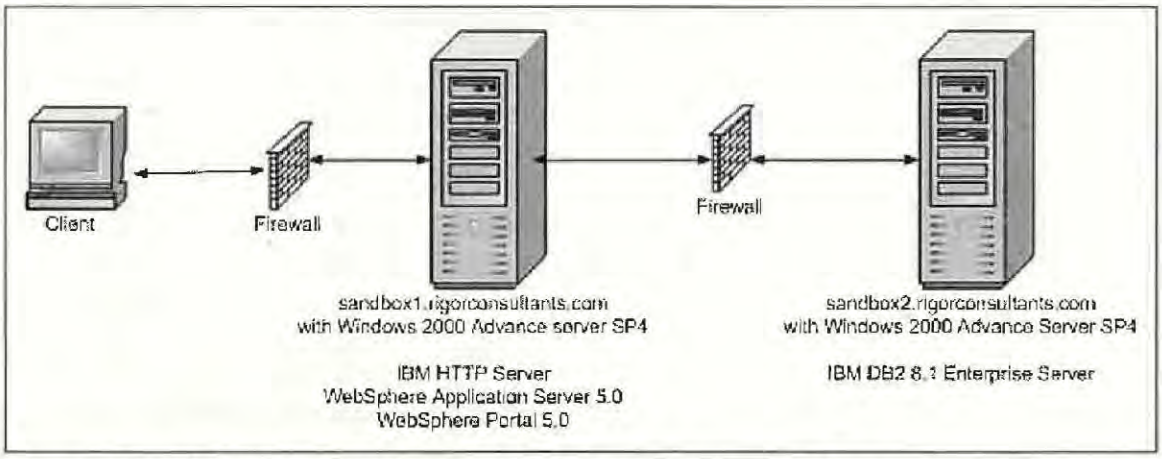

Figure 4-1 WebSphere Portal using DB2 database on a separate server.

management. Also by placing the database on its server, you can prevent the application server from impacting the database server and vice versa.

Configuring the DB2 Server

The first step is to configure the remote DB2 Server, which for the example is `sandbox2.rigoroconsultants.com`. Assume the following:

- You have installed DB2 UDB 8.1.
- The database user wpsdbusr has been set up and has administrative rights.
- You are logged in as wpsdbusr.
- Port 50000 and 50001 can access the client. This implies that the firewalls are configured to allow these ports.
- A db2 instance is available with enough storage allocated for the defined databases.

Three databases need to be created: wps50, wpcp50, and fdbk50. The database wps50 stores information about user customization and member management. Size is dependent on the number of portal objects and users. Database wpcp50 is used by the content publishing and the document management and the storage requirements are based on the number and size of documents being published or stored. The last database fdbk50 is the WPCP feedback database whose size is based on the site traffic.

Now you are ready to configure your remote server. This is somewhat complicated. Many of these commands are in the installation manual so a good trick is to find the installation manual (wpf-install-ent) in pdf format and copy and paste the commands from the manual to a command prompt.

The following are the steps to configure the remote database:

1. Log in to the DB2 server as wpsdbusr.
2. Enter the following commands in a DB2 Command Window to set up the WebSphere Portal database:

```
db2 create database wps50 using codeset UTF-8 territory us
db2 update database configuration for wps50 using applheapsz 16384
app_ctl_heap_sz 8192
db2 update database configuration for wps50 using stmtheap 60000
db2 update database configuration for wps50 using locklist 400
db2 update database configuration for wps50 using indexrec RESTART
db2 update database configuration for wps50 using logfilsiz 1000
db2 update database configuration for wps50 using logprimary 12
db2 update database configuration for wps50 using logsecond 10
db2set DB2_RR_TO_RS=yes
```

3. Enter the following commands in a DB2 Command Window to set up WebSphere Portal content publishing database:

```
db2 create database wpcp50 using codeset UTF-8 territory us collate
using identity
db2 create database fdbk50 using codeset UTF-8 territory us collate
using identity
db2 update database configuration for wpcp50 using applheapsz 4096
db2 update database configuration for fdbk50 using applheapsz 4096
db2 update database configuration for wpcp50 using logfilsiz 4096
db2 update database configuration for fdbk50 using logfilsiz 4096
db2 update database configuration for wpcp50 using logprimary 4
db2 update database configuration for fdbk50 using logprimary 4
db2 update database configuration for wpcp50 using logsecond 25
db2 update database configuration for fdbk50 using logsecond 25
```

4. Edit `C:\WINNT\system32\drivers\etc\services` and ensure the following entries for the DB2 connection and interrupt service ports:

```
db2c_db2       50000/tcp    # DB2 connection service port
db2c_db2CTLSV 50001/tcp    # DB2 interrupt service port
```

5. In a DB2 Command Window, set DB2COMM to TCP/IP and then set up the service name:

```
db2set DB2COMM=TCPIP
db2 UPDATE DBM CFG USING db2c_db2
```

6. Restart the database server.

Installing DB2 Client on WebSphere Portal Server

Next, you need to install the DB2 client on the WebSphere Portal Server so that it can access the remote DB2 server. This means you have to install the DB2 client and then configure it. Assume the following:

- WebSphere Portal, IBM HTTP Server, and WebSphere Administrator Server are on one machine and that it is a Windows 2000 Advance server.
- The user wpsdbusr has been set up and has administrative rights, is defined locally, can act as part of the operating system, and is able to create a token object, increase quotas, and replace a process level token.
- You are logged in as wpsdbusr.
- Port 50000 and 50001 can access the client. This implies that the firewalls are configured to allow these ports.

The following are the steps to do this:

1. Place the DB2 Enterprise Edition for windows CD into the drive. If autorun is enabled, you will see the Welcome screen. If not enter <D> **:Setup,** where <D> is your CD drive.
2. Click Install Products.
3. On Select the Product You Like to Install, click Next.
4. The Welcome to DB2 Setup screen appears. Click Next.
5. Accept the license agreement and click Next.
6. Choose Custom as the installation type and click Next.
7. Keep the default installation action and click Next.
8. The next screen will list the features you want to install. Unclick all features except Client Support. Also choose the installation directory. Click Next. See Figure 4-2.
9. A warning screen will appear if you have a firewall enabled. Click OK; however, it is strongly recommended to install the product with the firewall disabled.

Figure 4-2 DB2 features required on WebSphere Portal Server.

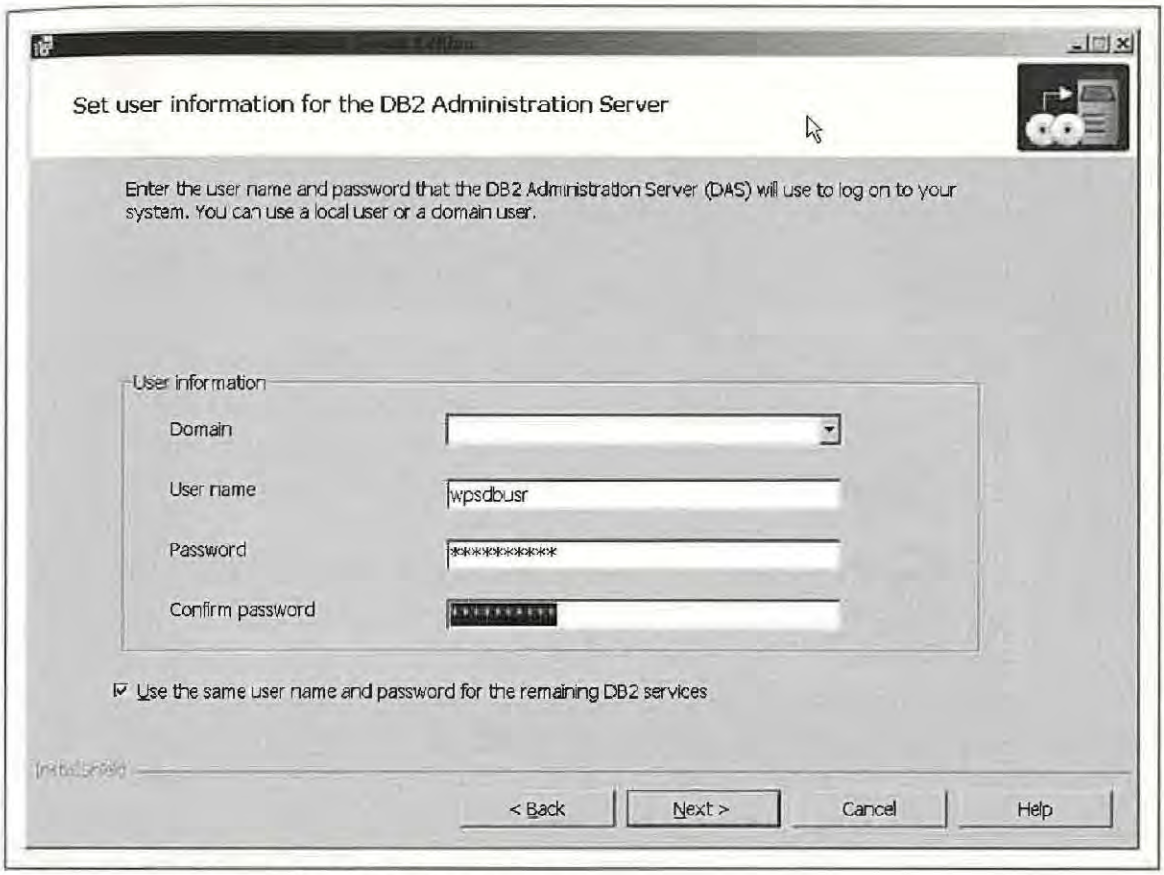

Figure 4-3 Defining the DB2 administrator.

10. Select installation language and confirm folder. Click Next.
11. Input user information for DB2 administrator and click Next. We have defined the DB2 administrator to be wpsdbusr. See Figure 4-3.
12. Set up your administrator contact list as per your company policies and click Next.
13. Keep the default Create a DB2 Instance and click Next.
14. Under Configure DB2 Instances, keep the default values and click Next. The name of the DB2 Instance on the client must be the same as the name of the DB2 Instance on the server.
15. Under Prepare the DB2 Tools catalog, keep the default value and click Next.
16. Specify your health contact as per your company policies and click Next.
17. Under Request Satellite Information, specify a satellite ID. This field is optional. If not specified, your login ID or wpsdbusr will be used.
18. Next will appear a screen listing the components you will be installing (for example, Figure 4-4). Click Install.

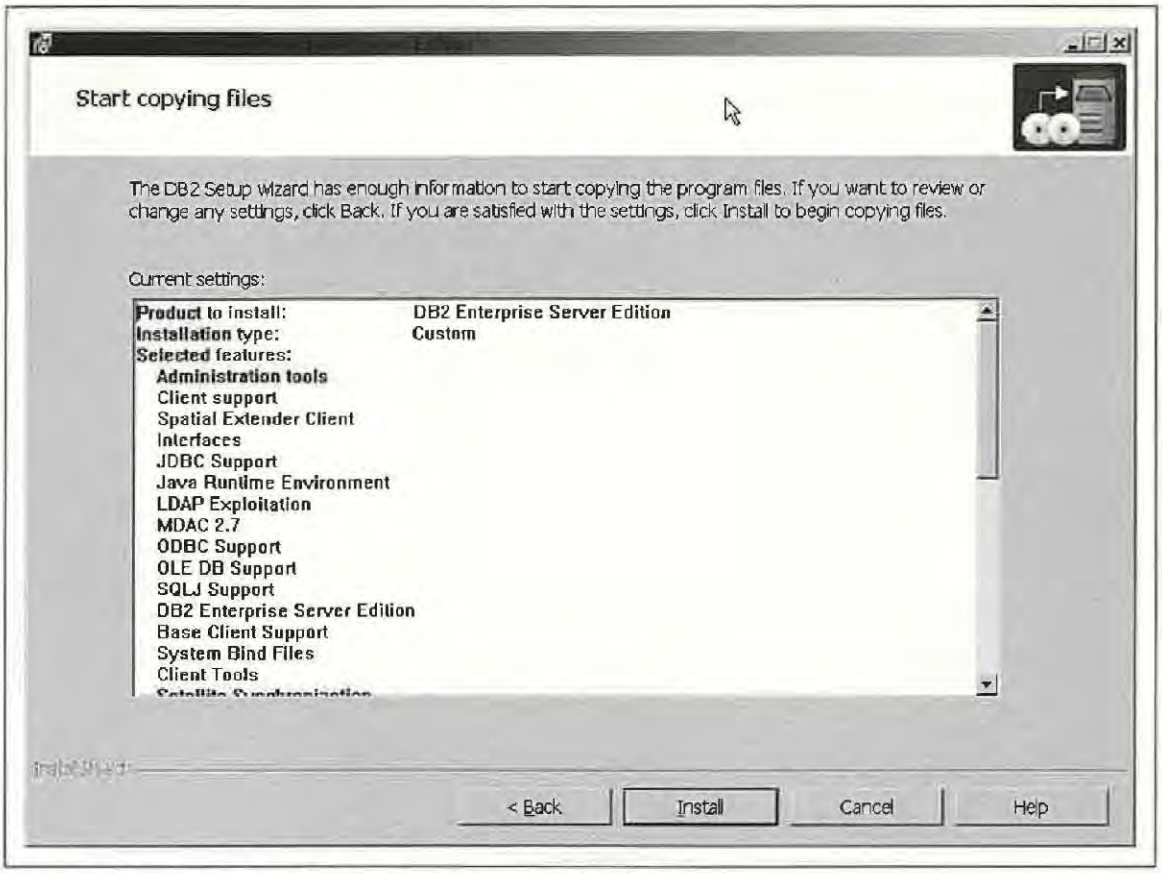

Figure 4-4 DB2 Client installation component list.

19. After you have successfully installed the client and restarted the server, log in as wpsdbusr.
20. In a DB2 Command Window, set DB2COMM to TCP/IP;

```
db2set DB2COMM=tcpip
```

21. In a DB2 Command Window, catalog the TCP/IP node with the IP address of the remote database;

```
db2 catalog tcpip node was1 remote sandbox2.rigoroconsultants.com
server db2c_db2
```

Here was1 is the node name we are assigning to the WebSphere Application Server node, `sandbox2.rigorconsultants.com` is our remote database server host name, and dbc_db2 is the name of the DB2 connection service port.

22. In a DB2 Command Window, catalog WebSphere Portal Server databases as aliases on the client.

```
db2 catalog db wps50 as wps50 at node was1
db2 catalog db wpcp50 as wpcp50 at node was1
db2 catalog db fdbk50 as fdbk50 at node was1
```

In this particular example, the name of the database on the server is the same as the alias on the client.

Using Oracle and SQL Server with WebSphere Portal

After you have installed DB2 on the client and server and created the appropriate tables, the next step is to modify WebSphere Portal configuration to support the database. But first let us go on a bit of a tangent to examine the support of other databases, since the last step is pretty independent of the database.

Although DB2 is an excellent database and comes with WebSphere Portal Extend and WebSphere Portal Experience, many sites have standardized on other databases. Two most popular databases are Microsoft's SQL Server and Oracle's database. In this section, you will find out what you need to do if you want to use SQL Server 2000 or Oracle 9i as the database instead of DB2 for the configuration in Figure 4-1. For the purpose of conciseness, let's assume that the databases have already been installed on the remote server and configured and tested to communicate with the WebSphere Portal Server. For Oracle, this implies that you are using the Oracle JDBC/OCI Type 2 drivers and that you copied the `classes12.zip` file from the remote Oracle server to the WebSphere Portal Server.

Step 1: Create the Databases

On `sandbox2.rigorconsultants.com`, create the databases wps50, wpcp50, and fdbk50. For SQL Server 2000, insure that your Collation Name is a case-sensitive collation. For English, set the collation name to SQL_latin1_General_CP1_CS_AS. For Oracle 8i, insure that the databases are created as UTF-8 character set databases.

Step 2: Create the Database Users

To use Oracle 9i or SQL Server 2000 with WebSphere Portal, we recommend six users be created in uppercase format: WPSDBUSR and WMMDBUSR that are associated with the database wps50, PZNADMIN, EJB, and WCMDBADM that are associated with wpcp50, and FEEDBACK that is associated with fdbk50. WPSDBUSR will own approximately 76 tables, which will consist of WebSphere Portal core objects. WMMDBUSR will own approximately 17 pages consisting of `Member Manager` objects. PZNADMIN

will own approximately 11 tables and EJB will own approximately 2 tables and they will both jointly own the tables for personalization information. WCMDBADM will own approximately 61 tables consisting of content publishing information. Lastly FEEDBACK will own 48 tables for the purpose of logging site and personalization usage. Note again this table size might fluctuate greatly depending on site usage, so configure it carefully.

Step 2a: Create SQL Server 2000 Database Users

To create database users for SQL Server, use SQL Server Enterprise Manager as follows:

1. Right-click SQL Server Group ⇨ Local ⇨ Security ⇨ Login and left-click on New Login.
2. Enter the database username **(WPSDBUSR, WMMDBUSR, PZNADMIN, EJB, WCMDBADM, or FEEDBACK)**.
3. Select SQL Server Authentication.
4. Set a password for the selected user.
5. Select the default database for each user using the mappings discussed in the beginning of this section. For instance, WPSDBUSR default database is wps50.
6. Click on the Server Role tab and make sure that no Server Role is selected.
7. Click on the Database Access tab. Enable permissions based on what's shown in Table 4-3.
8. Click OK to save the user changes.

Table 4-3 Recommended Database Access Roles

USERNAME	DATABASE	DATABASE ROLE
WPSDBUSR	wps50	public, db_owner
WMMDBUSR	wps50	public, db_owner
PZNADMIN	wpcp50	public, db_owner
EJB	wpcp50	public, db_owner
WCMDBADM	wpcp50	public, db_owner
FEEDBACK	fdbk50	public, db_owner

Step 2b: Create Oracle Database Users

To create the database users for Oracle, use SQL*Plus:

1. Enter username: **system/manager@wps50.** This will log the administrative user system with a password of manager into the wps50 database. Replace manager with the System Administrator's password.
2. Create the WebSphere Portal users and grant them appropriate privileges.

```
create user WPSDBUSR identified by PASSWORD default tablespace USERS
temporary tablespace TEMP;
create user WMMDBUSR identified by PASSWORD default tablespace USERS
temporary tablespace TEMP;
grant connect, resource to WPSDBUSR;
grant connect, resource to WMMDBUSR;
```

3. Connect to the content publishing database by entering **connect** and then username: **system/manager@wpcp50.** This will log the administrative user system with a password of manager into the wpcp50 database. Replace manager with the System Administrator's password.
4. Create the content publishing users and grant them appropriate privileges.

```
create user PZNADMIN identified by PASSWORD default tablespace USERS
temporary tablespace TEMP;
create user EJB identified by PASSWORD default tablespace USERS
temporary tablespace TEMP;
create user WCMDBADM identified by PASSWORD default tablespace USERS
temporary tablespace TEMP;
grant connect, resource to PZNADMIN;
grant connect, resource to EJB;
grant connect, resource to WCMDBADM;
grant insert any table to WCMDBADM;
```

5. Connect to the Feedback database by entering **connect** and then username: **system/manager@fdbk50**. This will log the administrative user system with a password of manager into the fdbk50 database and again replace manager with the System Administrator's password.
6. Create the content publishing feedback user and grant it appropriate privileges.

```
create user FEEDBACK identified by PASSWORD default tablespace USERS
temporary tablespace TEMP;
grant connect, resource to FEEDBACK;
```

7. Type **exit.**

Configuring WebSphere Portal to Access a Remote Database

Now that the database is installed and configured, you need to configure WebSphere Portal to access and populate the database with WebSphere Portal system and user data. The method used, as stated previously, will be the same for any remote database, except that the property values will be different. The configuration for a local database is simpler, because configuration templates exist for Oracle, SQL server, Informix, and DB2 that configure and transfer the data all in one step.

However for a remote database, such as the one shown in Figure 4-1, the process is more difficult. Perform the following steps to configure a remote database for WebSphere Portal on a Windows 2000 Advanced Server:

1. Open a command prompt and change the directory to `<wp_root>\config`.
2. Export the current database data by entering **WPSconfig.bat database-transfer-export**. Execution of this command can take some time; if execution is successful, a message will be displayed indicating success. You can also see if it has executed successfully by looking at the logs at `<wps_root>\log\configmessages`. Executing the command is similar under Unix except for the OS syntax differences. However in Linux the task name is database-transfer-export-linux. After the command has executed, all the data will be stored in the directory `<wps_root>\config\DBTRansfer`.
3. Copy `<wp_root>\config\wpconfig.properties` to `<wp_root>\config\wpconfig.bkp`.
4. Using a test editor, edit `<wp_root>\config\wpconfig.properties` and for the properties described in Tables 4-4–4-6, enter the appropriate values. The values entered in the table support the example defined in Figure 4-1. Note that this is a Java properties file, so a backward slash (\) is represented as a forward slash (/).
5. Save the file.
6. Test that the properties are valid by entering the following commands in a command prompt:

   ```
   WPSconfig.bat validate-database-connection-wps
   WPSconfig.bat validate-database-connection-wmm
   WPSconfig.bat validate-database-connection-wpcp
   ```

   These commands will test the database connection. If the response is positive go to the next step, otherwise check your properties to see if they are valid. Very often it is an invalid port or a port/IP address that is inaccessible due to a firewall configuration.

Table 4-4 Configuration Database Properties

PROPERTY	DESCRIPTION	PROPERTY VALUE		
		DB2	ORACLE	SQL SERVER
DbSafeMode	If this property is set to true, no database-specific updates are performed	False	False	False
DbType	Database used to store information for WebSphere Portal	db2	oracle	sqlserver
WpsDbName	Database name or alias name if remote where WebSphere Portal objects are stored	wps50	wps50	wps50
DbDriver	Name of the JDBC provider used to import SQL files	COM.ibm.db2. jdbc.app.DB2 Driver	oracle.jdbc.driver. OracleDriver	com.microsoft. jdbc.sqlserver. SQLServerDriver
DbDriverDs	The data source for the JDBC provider that WebSphere Portal used to import SQL files	Not applicable	oracle.jdbc.pool. OracleConnection-PoolDataSource	com.microsoft. jdbcx.sqlserver. SQLServerData Source
DbUrl	The database URL used to access the WebSphere Portal database	jdbc:db2:wps 50	jdbc:oracle:thin:@ sandbox2 .rigorconsulta nts.com:1521: wps50	jdbc:microsoft: sqlserver:// sandbox2 .rigorconsultants .com:1433:Database Name = wps50
DbUser	The database administrative username	wpsdbuser	WPSDBUSER	WPSDBUSER
DbPassword	The 1database administrative username password	–	–	–

(continued)

Table 4-4 *(continued)*

PROPERTY	DESCRIPTION	PROPERTY VALUE		
		DB2	ORACLE	SQL SERVER

DbLibrary	Location `db2java.zip` for DB2	`C:/Program Files/SQLLIB/java/db2/java.zip`	`C:/Program Files/oracle/jdbc/lib/classes12.zip`	`C:/Program Files/mssql/lib/mssqlserver.jar;`
				`C:/Program Files/mssql/lib/msbase.jar;`
				`C:/Program Files/mssql/lib/msutil.jar`

WpsDsName	The name of the datasource used for the WebSphere Portal database	wps50DS	wps50DS	wps50DS
WpsXDbName	The TCP/IP alias for the database to be used as dsname. Only needs to be set for Unix or Linux servers	wps5TCP	Not applicable	Not applicable
WpsDbNode	The node name of the WebSphere Portal database. Only needs to be set for Unix or Linux servers	wpsNode	Not applicable	Not applicable

Table 4-5 Configuration Content Publishing Database Properties

PROPERTY	DESCRIPTION	PROPERTY VALUE		
		DB2	ORACLE	SQL SERVER
WpcpDbHost Name	Host name of the machine running SQL Server	Not applicable	Not applicable	sandbox2.rigor-consultants.com
WpcpDbNode	The node name of the WebSphere Portal content publishing database. Only needs to be set for Unix or Linux server	wpcpNode	Not applicable	Not applicable
WpcpXDbName	The TCP/IP alias for the WebSphere Portal content publishing database. Only needs to be set for Unix or Linux server	wpcp5TCP	Not applicable	Not applicable
FeedbackXDb Name	The TCP/IP alias for the feedback database. Only needs to be set for Unix or Linux server	fdbk5TCP	Not applicable	Not applicable
WpcpDbName	The database name or alias name where the WebSphere Portal content publishing objects are created	wpcp50	wpcp50	wpcp50
WpcpDbUser	The database administrative username	wpsdbusr	WCMDBADM	WCMDBADM
WpcpDb Password	The password for the database user associated with WpcpDbUser			
WpcpDbUrl	The database URL used to access the WebSphere Portal content publishing database	jdbc:db2:wpcp50	jdbc:oracle:thin: @sandbox2 .rigorconsultants .com:1521:wpcp50	jdbc:microsoft:sql-server: // sandbox2 .rigorconsultants. com:1433:Database Name = wpcp50

(continued)

Table 4-5 *(continued)*

PROPERTY	DESCRIPTION	PROPERTY VALUE		
		DB2	ORACLE	SQL SERVER
WpcpDbEjb Password	The password for the Enterprise Java Bean (EJB) user	Not applicable		
WpcpDbPznad-min Password	The password for the PZNADMIN user	Not applicable		
FeedbackDb HostName	host name of the machine	Not applicable	Not applicable	`sandbox2.rigor-consultants.com`
FeedbackDb Name	The database name where the feedback objects are created	fdbk50	fdbk50	fdbk50
FeedbackDb User	The database administrative username	wpsdbusr	FEEDBACK	FEEDBACK
FeedbackDb Password	The password for the database user associated with FeedbackDbUser			
FeedbackDbUrl	The database URL used to access the feedback database	jdbc:db2:fdbk50	jdbc:oracle:thin:@ `sandbox2.rigor-consultants.com` :1521:fdbk50	jdbc:microsoft:sql-server://`sandbox2.rigorconsultants.com`:1433;Database Name = fdbk50

Table 4-6 Configuration Member Manager Properties

PROPERTY	DESCRIPTION	PROPERTY VALUE		
		DB2	ORACLE	SQL SERVER
WmmDsName	Name of the datasource to be used for the Member Manager database	wmmDS	wmmmDS	wmmmDS
WmmDbName	The database name or alias name where the Member Manager objects are created	wps50	wps50	wps50
WmmDbUser	The database administrative username	wpsdbusr	WMMDBUSR	WMMDBUSR
WmmDb Password	The password for the database user associated with WmmDbUser			
WmmDbUrl	The database URL used to access the Member Manager database	jdbc:db2:wps50	jdbc:oracle:thin:@ sandbox2 .rigorconsultants .com:1521:wps50	jdbc:microsoft:sql-server://sandbox2 .rigorconsultants .com:1433;Database Name=wps50

```
     [exec] Done Exporting DB Item 202
     [exec] Done Exporting DB Item 203
     [exec] Done Exporting DB Item 204
     [exec] Done Exporting DB Item 205
     [exec] Done Exporting DB Item 206
     [exec] Done Exporting DB Item 207
     [exec] Done Exporting DB Item 208
     [exec] Done Exporting DB Item 209
     [exec] Done Exporting DB Item 210
     [exec] Done Exporting DB Item 211
     [exec] Done Exporting DB Item 212
     [exec] Done Exporting DB Item 213
     [exec] Done Exporting DB Item 214
     [exec] Done Exporting DB Item 215
     [exec] Done Exporting DB Item 216
     [exec] Done Exporting DB Item 217

action-finalize-config:

action-propogate-properties:

BUILD SUCCESSFUL

Total time: 8 minutes 27 seconds
C:\Program Files\WebSphere\PortalServer\config>
```

Figure 4-5 Successful database configuration.

7. Enter the following command to import your old database data into your new database:

   ```
   WPSConfig.bat database-transfer-import
   ```

   After some time, the length being dependent on the volume of data imported, you should see a success message as indicated in Figure 4-5.

8. Restart WebSphere Portal and verify the installation by accessing your portal.

For DB2, you should perform a reorg check on all the database tables and if you installed WebSphere Portal on a pre-existing instance of WebSphere Application Server, make sure that the path to the required DB2 JDBC provider files was added to the WebSphere Application Server. You can verify that the classpath has been set up correctly by doing the following steps:

1. Go to administration console.
2. Click Servers Application Servers.
3. Then click WebSphere Portal from the list of application servers.
4. Click Process Definition from the list of additional properties.
5. Finally, click Java Virtual Machine from the list of additional properties.

Summary

In this chapter you saw how to configure WebSphere Portal after a basic installation. We focused on enabling WebSphere Portal to store its objects in

DB2 8.1, Oracle 9i, or SQL Server 2000. Configuration of WebSphere Portal for LDAP will be covered in Chapter 19. You saw that with WSConfig utility, you could easily change your configuration without reinstalling the product and the configuration utility also handled data migration.

In the next chapter we will discuss how to install some specialized tools that come with WebSphere Portal and also introduce you to some tools and features that help you analyze installation and runtime problems.

CHAPTER

5

Installing WebSphere Portal Tools and Problem Analysis

WebSphere Portal comes with a couple of tools that are critical to the development and usage of portals: Portal Toolkit 5.0.2 and IBM Lotus Workplace Web Content Management version 1.1. In this chapter we will show you how to install them, since the installation methods are not that straightforward.

Lastly, we will discuss what you should do if you encounter a problem and show the tools that are available for identifying your problems and correcting them. This will include a discussion on Tivoli Web Site Analyzer that comes with WebSphere Portal Enable. The Tivoli Web Site Analyzer allows you to analyze your portal usage.

We will look at the 10 most common problems customers have encountered and propose some best practices.

Portal Toolkit 5.0.2

Portal Toolkit 5.0.2 is a plug-in for WebSphere Studio that enables you to easily develop and debug portlets and then test them either using an integrated local WebSphere Portal Test Environment or on a remote WebSphere Portal server. The local WebSphere Test Environment can be WP V4.2, WP 5.0, or WP 5.0.2. The usage of the Portal Toolkit is discussed in Chapter 15. Here we are going to discuss how to install Portal Toolkit 5.0.2 with a local WebSphere Portal 5.0.2 Test environment.

First we are assuming that you have installed WebSphere Studio Application Developer V5.1.0 or WebSphere Studio Site Developer V5.1.0. Do not

install WebSphere Studio 5.1.0. Interim Fix 001 if you need to use the Web Service Client Portlet Project Wizard. Also when you install WebSphere Studio, insure that the installation directory has a short path. The toolkit installation tends to have problems with long path names.

The Portal Toolkit has the following system and software requirements:

1. Processor: 800 MHz Pentium III minimum.
2. Memory: 1 GB of memory minimum.
3. Disk space: 2.5GB +.
4. Operating system: Windows 2000 SP2+ or Windows XP SP1. Include all security patches.
5. Browser: IE5.5 SP2 +.
6. Tools: Adobe Acrobat, Zip Utility.
7. WebSphere Studio Application Developer 5.1.0 or WebSphere Studio Site Developer 5.1.0.
8. WebSphere Portal Toolkit V5.0.2 MP from WebSphere Portal Zone for developers; `http://www-106.ibm.com/developerworks/websphere/zones/portal/`.
9. Download WebSphere Application Server 5.0.2 Fix Pack and the WebSphere Portal 5.0 Fix Pack 2, which can be found at `http://www-1.ibm.com/support/docview.wss?rs=0&q1=portal+5.0.2&uid=swg24006309&loc=en_US&cs=utf-8&cc=us&lang=en`.

Updating WebSphere Studio's WAS Runtime Environment

The first step is to update the WAS V5.0.2 runtime environment used by WebSphere Studio to enable local debugging for WP V5.0.2. You do this by taking the following steps:

1. Unzip WebSphere Application Server 5.0.2 Fix Pack into `c:\fixes`.
2. Open a command prompt, change directory to `C:\fixes\win` and set `JAVA_HOME=<WSAD>\runtimes\base_v5\java` and hit Enter.
3. Type **updateWizard.bat.** and hit Enter.
4. Using the drop-down box, select a language for the wizard to use.
5. Read the Welcome screen and click Next.
6. Verify that Specify Product Information is checked and then browse to `<WSAD>\runtimes\base_v5` and click Next.
7. Select Install Fixes and click Next.

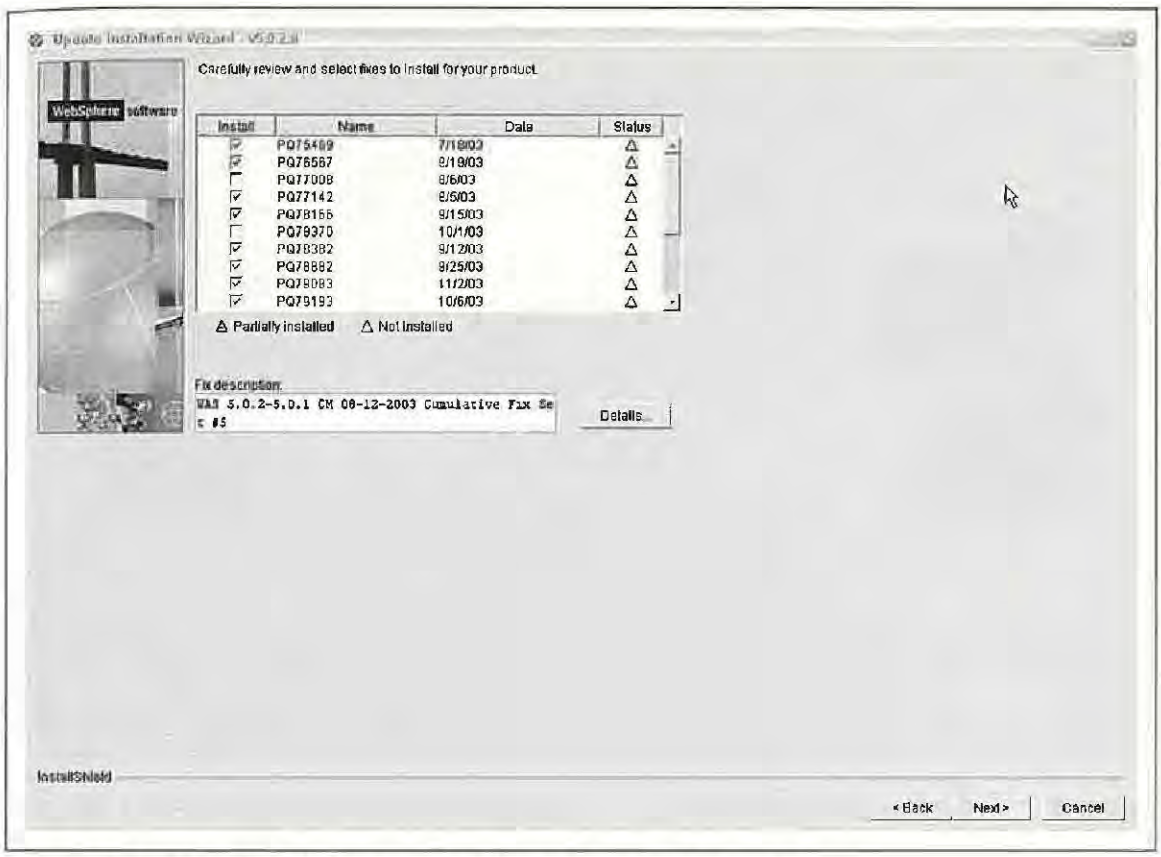

Figure 5-1 Fixes for WebSphere Studio runtime environment.

8. Verify/update the Fix directory to `C:\fixes\win\efixes` and click Next.
9. As shown in Figure 5-1, select all the fixes in the list except for PQ77008, PQ78370, and WAS_Plugin_09-03-2003_5.0.X_cumulative and then click Next. If you choose PQ77008, PQ78370, and/or WAS_Plugin_09-03-2003_5.0.X_cumulative, the installation will fail.
10. Review the fixes to be installed or refreshed and then click Next.
11. Check that the Update Wizard reports that the fixes have been installed successfully and then click Finish.

Installing WebSphere Portal Toolkit V5.0.2 MP

The next step is to install the actual Portal Toolkit with the WebSphere Test Environment V5.0. Perform the following steps to do this:

1. Shut down your HTTP servers.
2. Double-click `PortalToolkit502MP.exe`.
3. Enter a temporary directory and click Next.
4. Select a language for the wizard to use from the drop down box and click OK.

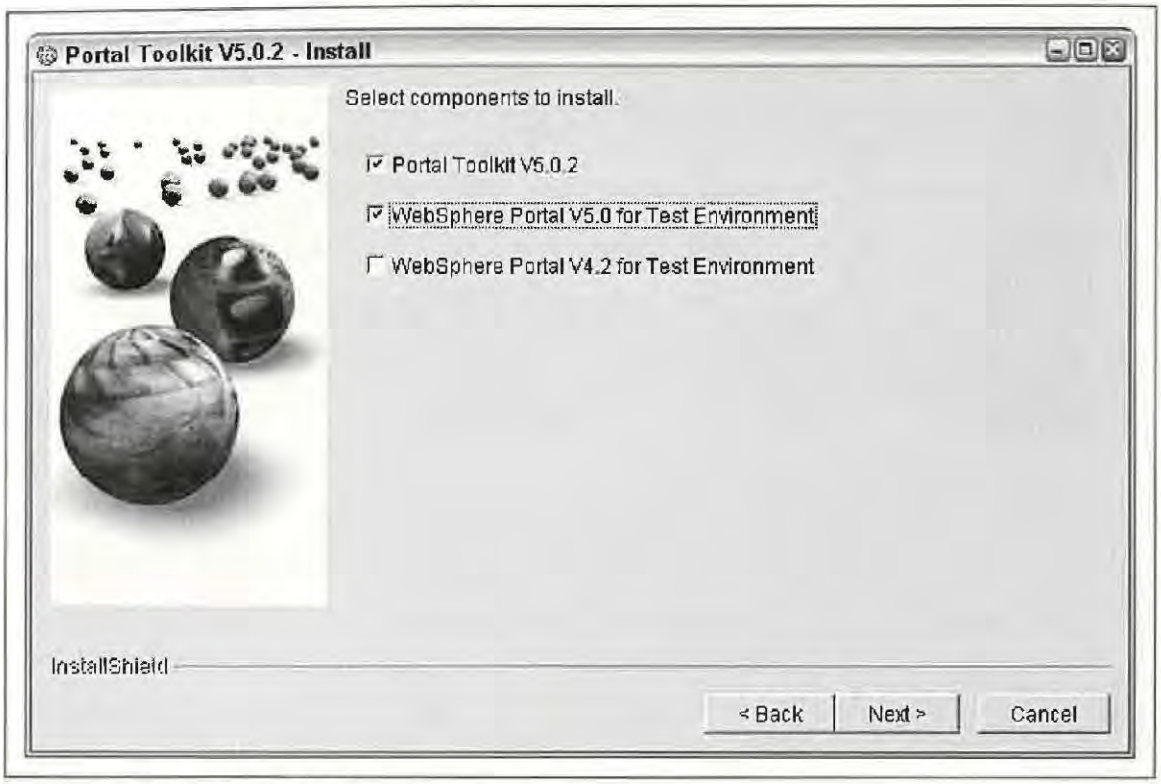

Figure 5-2 Portal Toolkit components.

5. Read the Welcome screen and click Next.
6. Read the License Agreement, select I Accept the Terms in the License Agreement, and click Next.
7. In the list of components to install (as seen in Figure 5-2), select Portal Toolkit V5.0.2 and WebSphere Portal V5.0 for Test environment. Click Next.
8. In the WebSphere Portal V5.0 installer field, place CD 2 of WebSphere Portal Enable or Extend. Browse to the location of `wpsinstaller.jar`, which is the wps directory (see Figure 5-3). Click Next.
9. Review the install location of the WebSphere Portal V5.0 for Test Environment and click Next.
10. Once the installation is completed successfully, click Finish. It will take 20–30 minutes.
11. Double-click the WebSphere Studio shortcut on the desktop.
12. In the WebSphere Studio dialog box, verify your workspace directory and click OK.
13. As seen in Figure 5-4, a Configuration Changes dialog should appear. Expand the detected changes and verify whether the Server

Figure 5-3 WebSphere Portal V5.0 location.

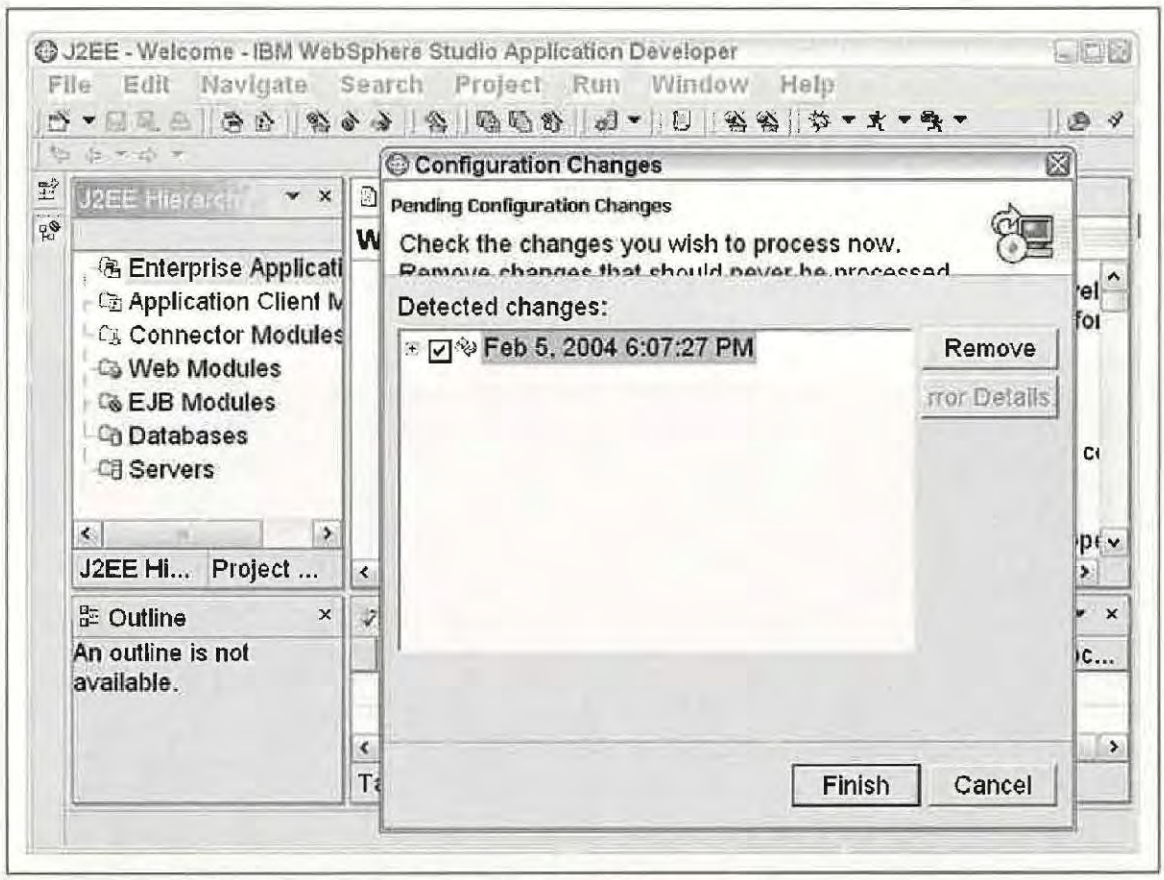

Figure 5-4 WebSphere Studio Configuration Changes dialog.

Tools for WebSphere Portal version 5.0 Test Environment 5.1.0 and WebSphere Portal Toolkit have been detected. Click Finish.

14. An Install/Update dialog appears and asks you to restart WSAD to incorporate the changes, click Yes.
15. After the restart, WSAD should display the basic development environment again.

Updating to WebSphere Portal Test Environment V5.0.2

Next you have to update the WebSphere Portal V5.0 Test Environment to V5.0.2. Do this by performing the following steps:

1. Extract WebSphere Portal Fix Pack 2 to `c:\fixpack2`. The location should be at the root path and have no blanks or special characters. The batch file will not work with a complex path. So make your life easier and use `c:\fixpack2`.
2. Open a command prompt and change directory to `<WSAD>\Portal Toolkit`.
3. Run updatePortalTestEnv502 `C:\fixpack2`. It should take approximately 20 minutes for the update to complete.
4. Check results in the command prompt (as seen in Figure 5-5) for `updatePortalTestEnv502.bat` to make sure it ended successfully.
5. Verify the WebSphere Portal V5.0.2 Test Environment upgrade by doing the following.

Figure 5-5 A successful update of the Portal Test Environment.

a. Double-click the WebSphere Studio shortcut on the desktop.

b. In the WebSphere Studio dialog box, verify your workspace and then click OK.

c. Click File ⇨ New ⇨ Project ⇨ Portlet Development Portlet Application Project.

d. To define the Portlet Project, enter **Portlet1** for the Project Name and click Finish.

e. After the wizard finishes creating the project, right-click on Portlet1 in the Project Navigator and select Debug on Server.

f. In the Server Selection dialog, select WebSphere Portal V5.0 Test Environment under the Create a New Server Area. Check the Set Server as Project Default (do not prompt) and click Finish.

g. The project will be published to the WebSphere Portal V5.0 Test Environment server and the server will be started. Server startup information will be displayed in the Console view in the bottom center area.

h. After sometime, a browser should display a Debug page with Portlet1 portlet on it (as seen in Figure 5-6). This verifies that the Test Environment is running properly.

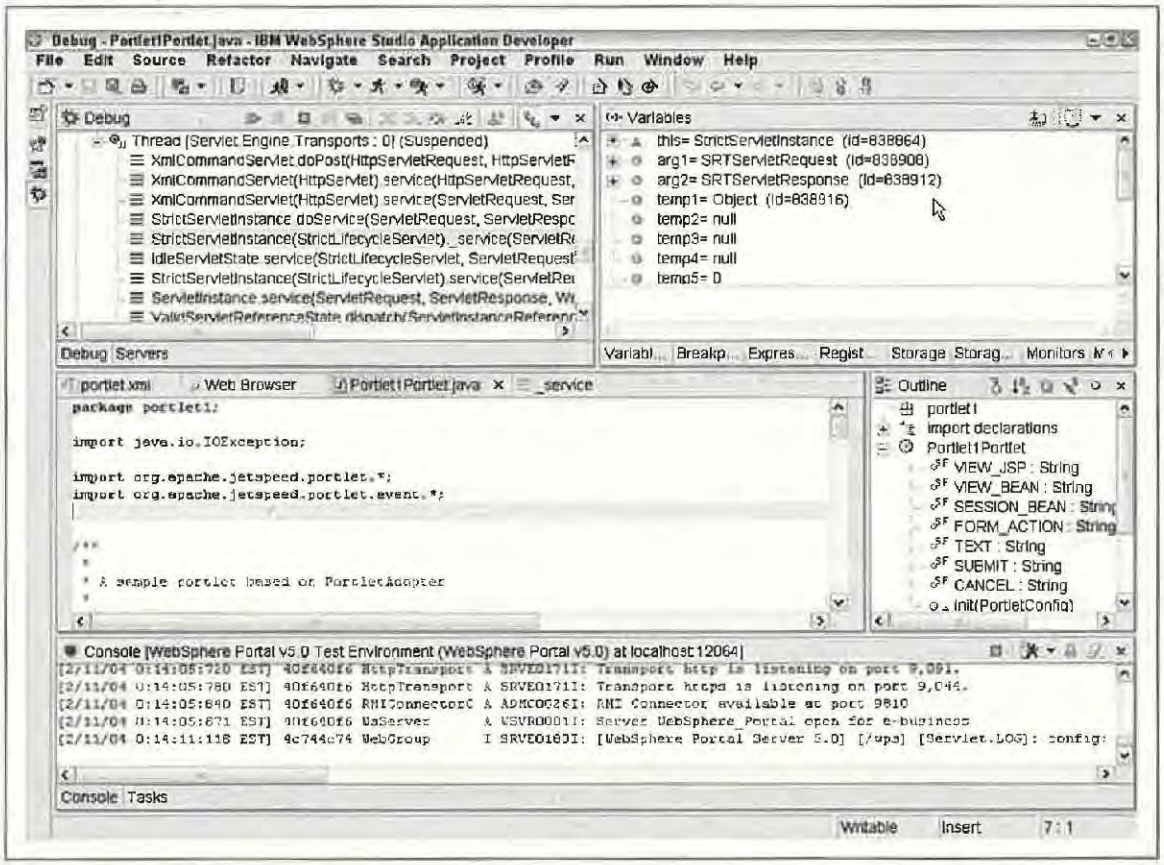

Figure 5-6 WebSphere Studio running Server in Debug mode.

IBM Lotus Workplace Web Content Management Version 1.1

Included with WebSphere Portal V5.0.2 is IBM Lotus Workplace Web Content Management (ILWWCM) version 1.1, which eventually replaces WPCP. However, this component must be installed after you have installed WebSphere Portal.

On a separate CD you will find a file called `C53YLML.exe`. You will also need to download `JRE 1.3.1_09` and install it. To install ILWWCM, perform the following steps.

Updating the Initialization Parameters

1. Log in to Windows as an Administrative user.
2. Double-click `C53YLML.exe` and unzip the files to `c:\wcm`.
3. Open the WebSphere Application Assembly Tool by clicking Start ⇨ Programs ⇨ IBM WebSphere ⇨ Application Server V5.0 ⇨ Application Assembly Tool.
4. In Welcome to the Application Assembly Tool, select the Existing tab.
5. Browse to `C:\wcm\app\config\websphere\ilwwcm.war`. Click Select and then click OK.
6. Navigate to and select Initialization Parameters by clicking ILWWCM ⇨ Web Components ⇨ ILWWCM Framework ⇨ Initialization Parameters.
7. As seen in Figure 5-7, replace parameter `[ILWWCM_HOME]` with `C:/wcm/app` and click Apply. Click Save and exit the Application Assembly Tool.

Installing the ILWWCM Web Application

1. Start WebSphere Application Server.
2. Open the WebSphere Administration Console.
3. Select Applications ⇨ Install New Applications.
4. Select Local Path, browse to `C:\wcm\app\config\websphere\ilwwcm.war`, and click Open.
5. As seen in Figure 5-8, enter **wcm** in the Context Root field and then click Next.
6. On all the next four screens, accept the defaults and click Next until you see the last screen. Then hit Finish.

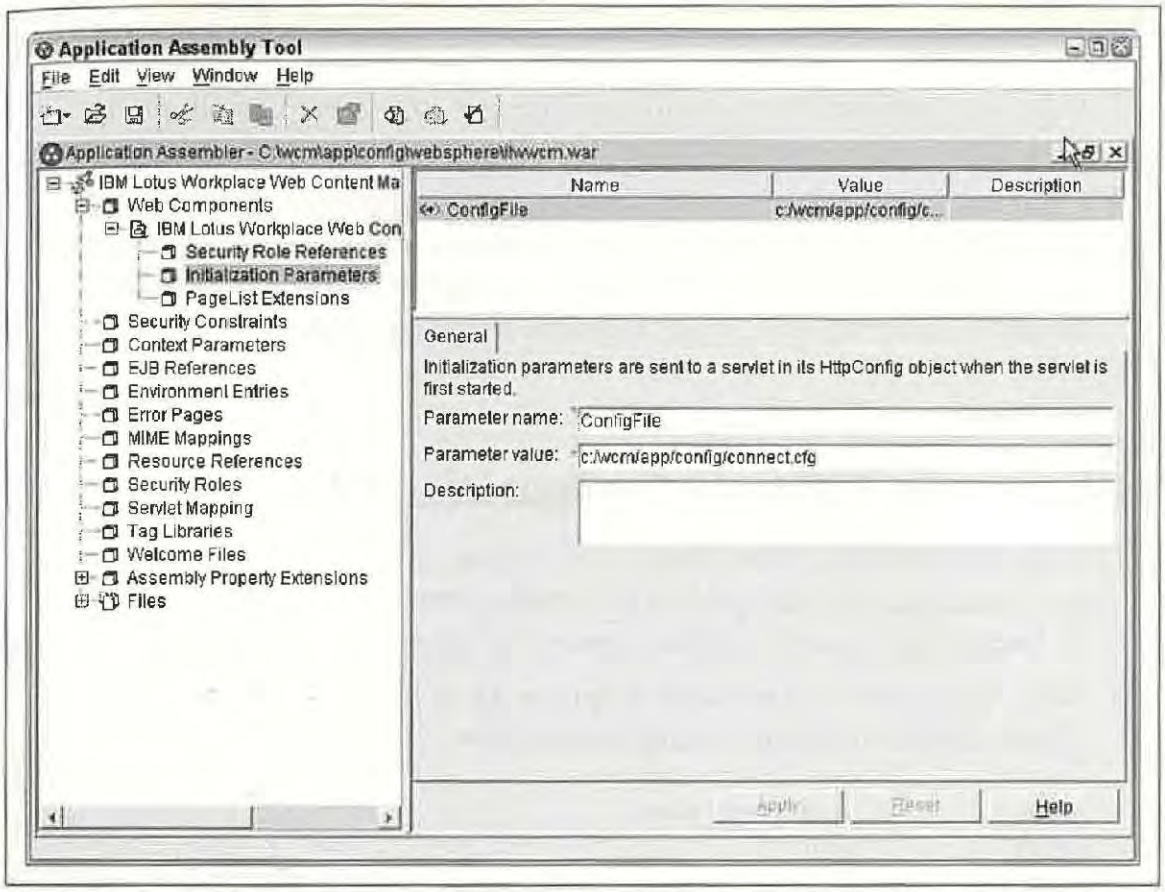

Figure 5-7 Configuring ILWWCM in WebSphere Application Assembly Tool.

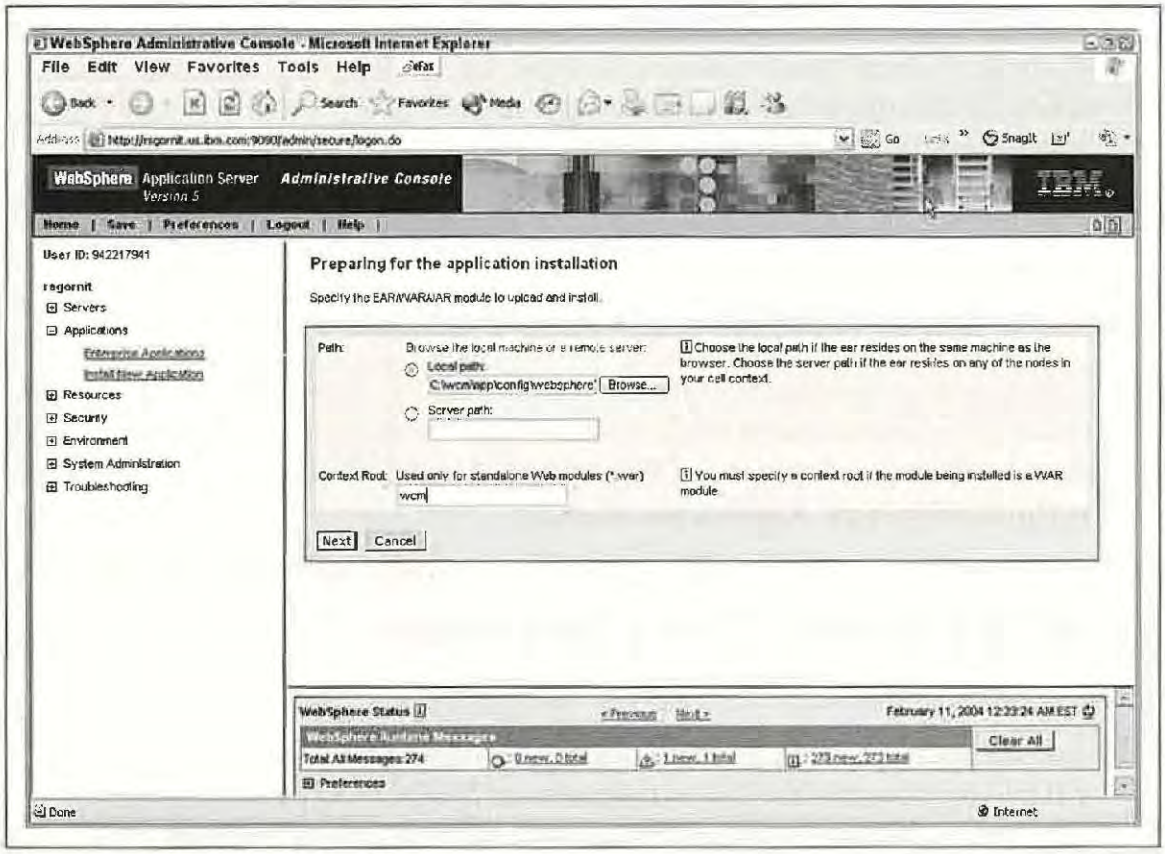

Figure 5-8 Installing ILWWCM.

7. If the installation is successful, click Save to Master Configuration and then in the Save to Master Configuration window click Save.
8. Copy the contents of `C:\wcm\web-app`, which consists of three directories (`ac_admin`, `WEB-INF`, and `webinterface`) to paste the directories into the `<WAS>\installedApps\<short_server_name>\ilwwcm_war.ear\ilwwcm.war` directory.

Updating the ILWWCM Configuration Files

1. Copy the configuration files `aptrixjpe.properties`, `aptrixsearch.properties`, and `connect.cfg` from `C:\wcm\app\config\samples` to `C:\wcm\app\config`.
2. Now this process is tedious. Replace the following variables with the given values in all the configuration files.

```
[HOST]         = Your host name.
[PORT]         = 9080
[CONTEXT_ROOT] = wcm
[ILWWCM_HOME]  = C:/wcm/app
[WEB_APP_HOME]= <WAS>/installedApps/<short_server_name>/
ilwwcm_war.ear/ilwwcm.war
```

   Note that since these files are Java property files, the backward slash used for Windows directory is represented as a forward slash. This is not a typo in the book.
3. Save each file.

Starting the ILWWCM Web Application

In the WebSphere Administration Console, find and select the `ilwwcm_war` Enterprise Application and click Start. Verify `ilwwcm_war` has started by checking for the green arrow in Enterprise Applications. You can also just restart the WebSphere Application Server.

ILWWCM Administrative Client Setup

1. Make sure that `JRE 1.3.1_09` is installed.
2. Initialize the ILWWCM Web application by entering **http://localhost:9080/ilwwcm/connect/?MOD=AJPEGUI** in an Internet Explorer browser. Figure 5-9 indicates a successful initialization.

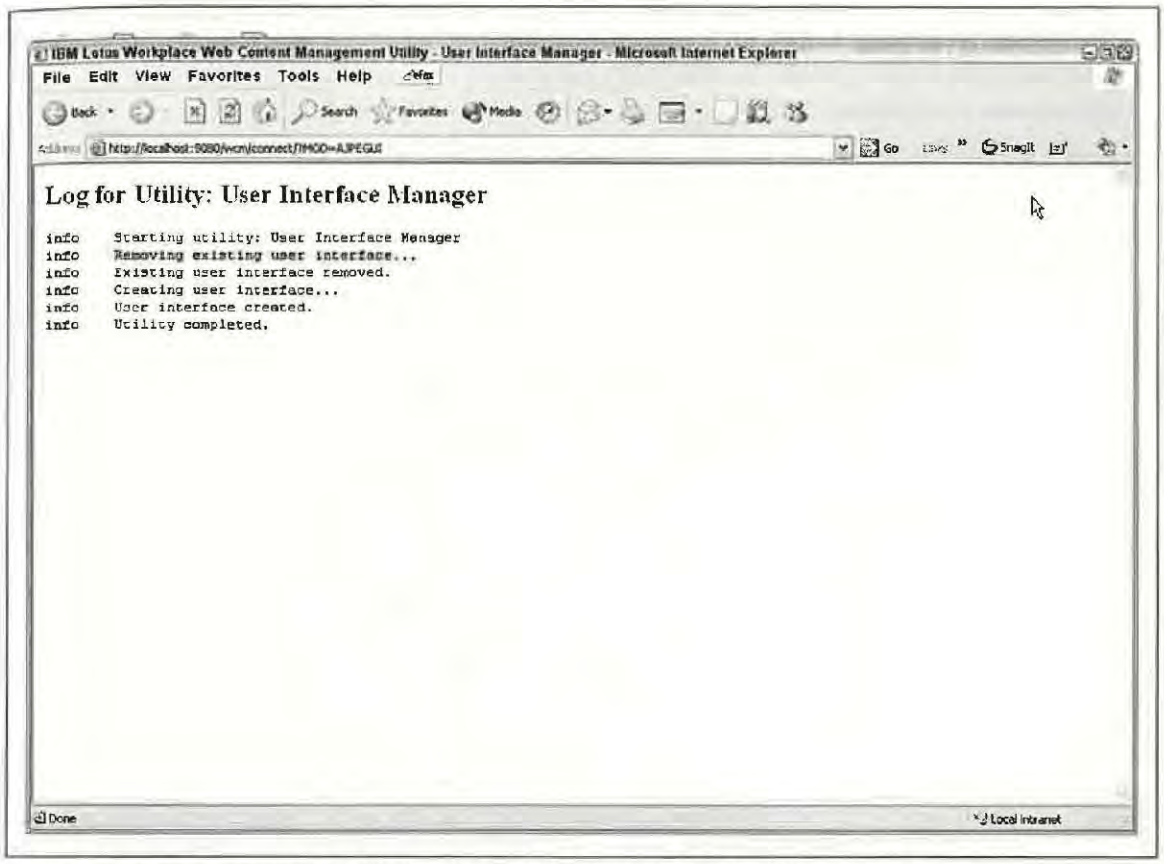

Figure 5-9 A successful initialization of the ILWWCM Administrative Client.

3. In Enter Network Password, enter **Administrator** for the User Name and password for Password. Verify that the Utility to update the User Interface Manager is completed without errors.
4. Enter **http://localhost:9080/ilwwcm/connect/aptrix** in a browser. In 'Java Plug-in Security Warning' click Grant Always to allow the Verisign authenticated ILWWCM applet to install and run. Expand the components of the Home area and verify that Aptrix exists as a Site Framework. If you see Figure 5-10, your ILWWCM administrator client is now running.

Installing the WebSphere Portal Content Management Portlet

To install the portlet, which will enable you to view your content managed by ILWWCM, log on WP as an administrator:

1. Go to the Portal Administration screen.
2. Go to the Install Portlets view and Click Browse.

Figure 5-10 The ILWWCM Administrator Client.

3. Select the `wpsV5-j-ilwwcm-cportlet.war` file to install and click Next. It is located at `c:\wcm\contentportlet\wpsV5`.
4. The Web Content Management Portlet is displayed (as seen in Figure 5-11). Click Install to install. The portlet will be labeled as Content Viewer, Java Edition when you look for it under Manage Portlets.

Problem Analysis

The one law that most professionals in the computer field believe in religiously is Murphy's law: whatever can go wrong will go wrong. This is why problem analysis is so important. You know something will go wrong. The question is how fast you can diagnose it, correct the situation, and prevent it from recurring.

WebSphere Portal is not a simple product. It is a framework that interacts with numerous other disparate products. Many problems occur with these interactions.

The first step to determine a problem is to isolate and simplify the environment. Try to duplicate problems in self-contained networks free of

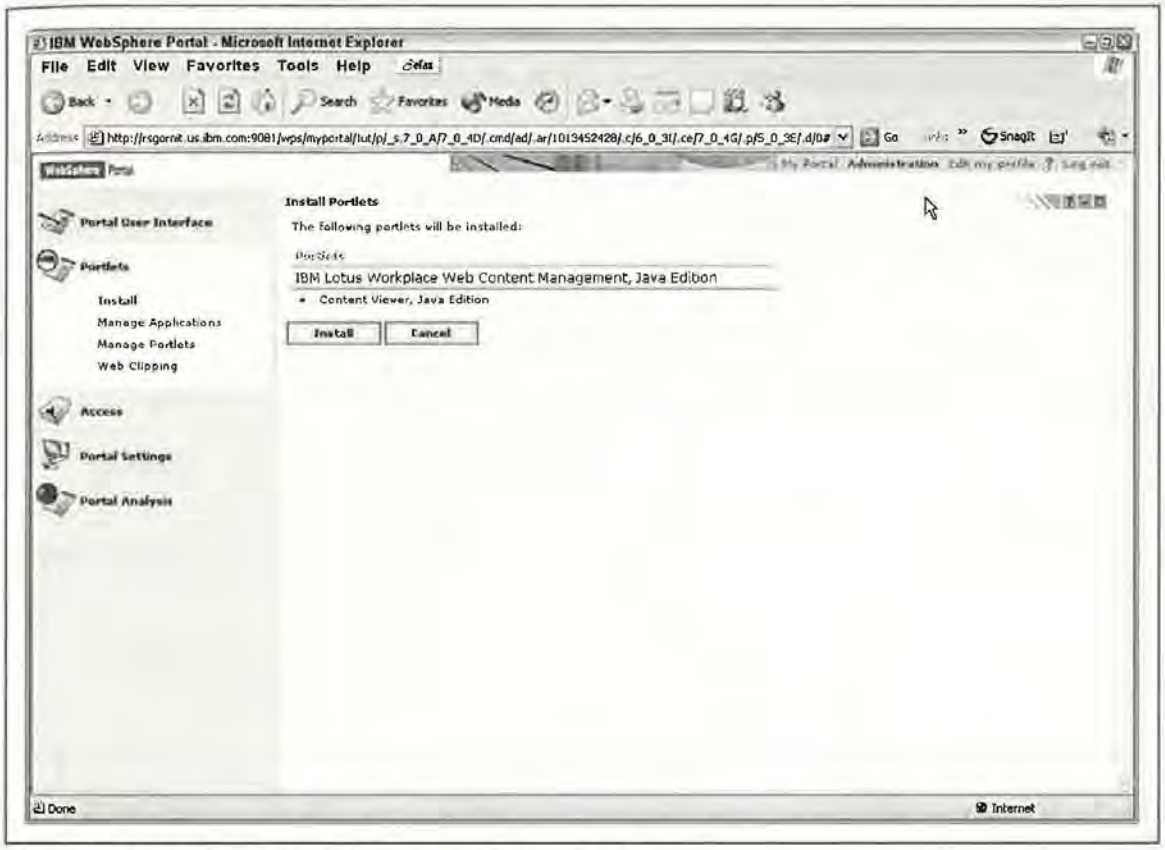

Figure 5-11 The ILWWCM Content Viewer Portlet.

firewalls and external HTTP servers. In other words, try to duplicate the error using WAS internal HTTP server on a single machine. When trying to isolate a problem, change only one parameter at a time. Check the infocenter and the WebSphere Portal developer Community Zone. The probability is that you are not the first to encounter the problem.

Try to find the errors before production by doing extensive load and stress testing. The time you spend on load testing will be one-fifth the time you will spend if you find the error during production (and a lot less stressful on yourself).

Remember, there is nothing magical about WebSphere Portal. It is a collection of applications that run under WebSphere Application Server and as such suffer the same problems as many other WebSphere Applications such as misconfigured HTTP plug-ins, memory, native, and application heap leaks. You can also suffer because of problems from misconfigured secured, classloaders, JVMs, and Session Managers. Problems can also be caused (most probably) by poorly written portlets.

WebSphere Portal helps you find your problems by logging everything. During the installation, log files are created in `<wp-root>\log`. However, before the Portal Server directory is created, the install logs are in the

Table 5-1 WebSphere Portal Installation Logs

LOG FILE NAME	DESCRIPTION
wpinstalllog.txt	Installation trace information
installmessages.txt	Installation messages that are generated during installation
wpcpInstallLog.txt	Trace information generated by WPCP
wpsinstalllog.txt	Trace information generated by WP
portletinstall.txt	Portlet installation messages
wpwasfp1.txt	WebSphere Application Server Fix Pack 1 trace information
wppmefp1.txt	WebSphere Application Server Enterprise Edition Fix Pack 1 trace information
log.txt	WebSphere Application Server trace information. Located in <was_root>/logs directory
WAS.PME.install.log	WebSphere Application Server Enterprise Edition trace information. Located in <was_root>/logs directory

system temporary directory. The logs created during installation are shown in Table 5-1.

Finding problems during WebSphere Portal provides extensive trace capability (see Chapter 10 for grammar and trace string configuration). Trace logging is set by enabling properties in the <wp_root>/shared/app/config/log.properties file. When the trace is set, it creates a file <wp-root>/log/wps_<date_time>.log. In the WebSphere Portal Administration guide under Using Logs, IBM lists in detail the trace settings needed to analyze if a component is having a problem.

Tivoli Web Site Analyzer

A useful tool to help you understand your WebSphere Portal site characteristics is Tivoli Web Site Analyzer. This tool comes with WebSphere Portal Enables and also comes with features preconfigured to help analyze WebSphere Portal.

Tivoli Web Site Analyzer analyses WebSphere Portal data that is stored in site analysis logs. You have to enable these logs by doing the following:

1. Open <wp-root>/shared/app/config/services/SiteAnalyzerLogService.properties with a text editor.

2. Accept the default values. Data will be stored in `log/sa.log` and the backup log will be `log/sa_$CREATE_TIME.log` where $CREATE_TIME is the time and date when the file was created.
3. Accept the default date and time format used as part of the log file name. The default value is yyyy.MM.dd-HH-mm.ss
4. Uncomment one of the following parameters to control the interval at which the log file is being backed up.
   - `SiteAnalyzerFileHandler.minutesPerLogFile` if you want to log in intervals of minutes, and set the value to an integer in the range 1–60.
   - `SiteAnalyzerFileHandler.hoursPerLogFile` if you want to log in intervals of hours, and set the value to an integer in the range 1–24.
   - `SiteAnalyzerFileHandler.daysPerLogFile` If you want to log in intervals of days, and set the value to an integer that indicates the number of days.
5. Enable the logger you want activated by removing the comment indicator (#). If you disable the logger, you re-add the comment indicator in front of it. Table 5-2 shows which WebSphere Portal URLs the loggers track.
6. Restart WebSphere Portal.

For purpose of demonstration, we enabled all the loggers, logged in as **wpsadmin**, and clicked on various pages and portlets. Remember, this is for demonstration. Normally you would not enable all the loggers (unless you are doing a major stress test on the box) and you would monitor more than one user.

Once enabled, WebSphere Portal started tracking the URL usuage as `sa.txt`. If you look in sa.txt, you would see entries like the following:

```
rsgornit.us.ibm.com - wpsadmin [10/Feb/2004:17:34:32 +0000] "GET
/Portlet/5_0_6B/YourCo_Financial_Latest_News_Portlet?PortletPID=5_0_6B&P
ortletMode=View&PortletState=Normal HTTP/1.1" 200 -1
"http://rsgornit.us.ibm.com/Page/6_0_6A/Home" "Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)" "JSES-SIONID=QUXWVc
_MeJHUly4BYAPd07u"
```

These entries show the following:

- The IP address or the domain name: `rsgornit rsgornit.us.ibm.com`
- The identifier used to identify the client. If it is not known then a dash (-) is used.

Table 5-2 Tivoli Web Site Analysis WebSphere Portal Loggers

PORTAL SERVER ANALYSIS LOGGER DESCRIPTION	URL ACTIVITY LOGGED
`SiteAnalyzerSessionLogger.isLogging`	/Command/Login /Command/Logout
`SiteAnalyzerUserManagementLogger.isLogging`	/Command/UserManagement/CreateUser /Command/UserManagement/DeleteUser /Command/UserManagement/CreateGroup /Command/UserManagement/DeleteGroup
`SiteAnalyzerPageLogger.isLogging`	/Page/* /Command/Customizer/CreatePage /Command/Customizer/EditPage /Command/Customizer/DeletePage
`SiteAnalyzerPortletLogger.isLogging`	/Portlet/*
`SiteAnalyzerPortletActionLogger.isLogging`	/PortletAction/*
`SiteAnalyzerErrorLogger.isLogging`	/Error/Portlet /Error/Page

- The WebSphere Portal user ID: `wpsadmin`.
- The date and time of the HTTP request: `[10/Feb/2004:17:34:32 +0000]`
- The HTTP request:

```
"GET
/Portlet/5_0_6B/YourCo_Financial_Latest_News_Portlet?PortletPID=5_0_6
B&PortletMode=View&PortletState=Normal HTTP/1.1"
```

- The HTTP status code: `200`.
- The number of bytes transferred from the client due to the request. -1 means unknown: `-1`.
- The URL that linked the client to the Web site:

```
"http://rsgornit.us.ibm.com/Page/6_0_6A/Home"
```

- The type of Web browser:

```
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR
1.1.4322)"
```

- The name and value of the cookie sent:

```
"JSESSIONID=QUXWVc_MeJHUly4BYAPd07u"
```

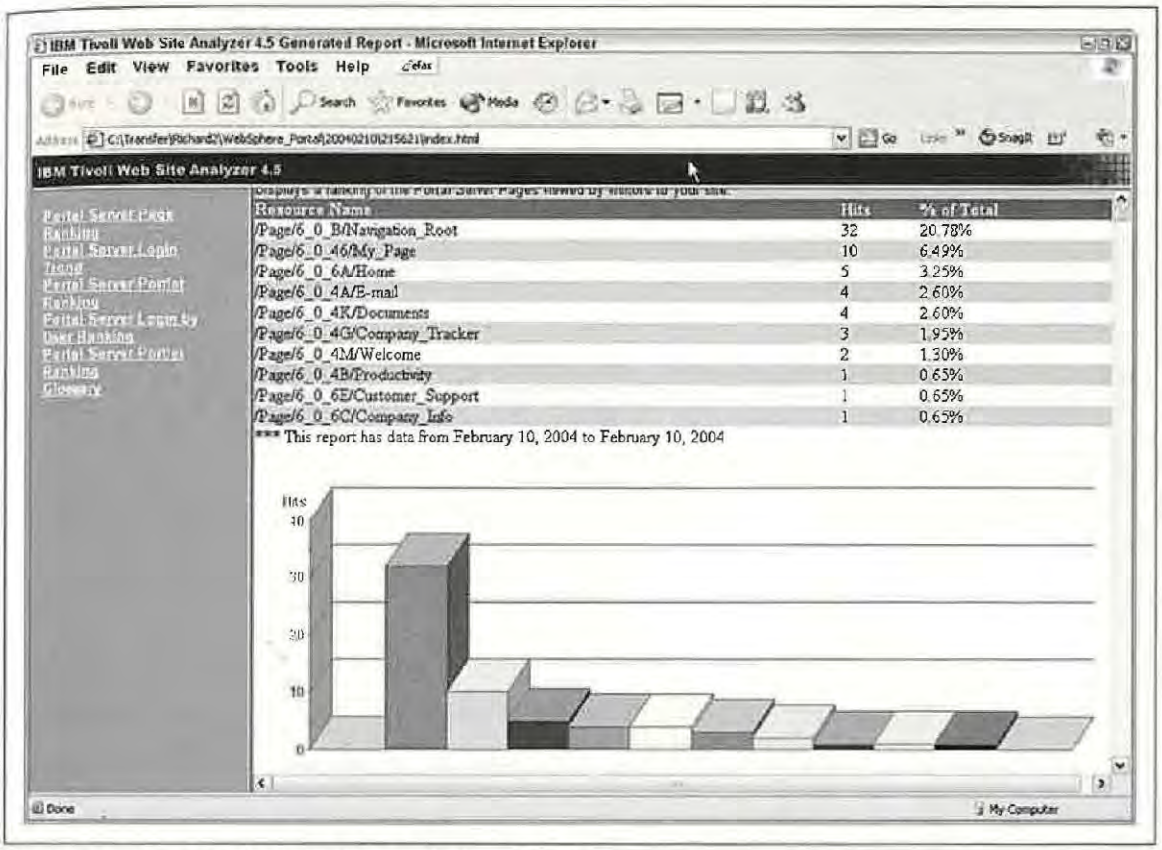

Figure 5-12 Tivoli Web Site analysis of the log.

You can then import your logs into Tivoli Web Site Analyzer or have it sample the logs automatically at scheduled intervals. Tivoli Web Site has special report elements defined for portal server. You can view pages, portlets, and logins rank by visitors. You can summarize statistics by commands or view hoe many times a page has been edited. Figure 5-12 shows how many times each page has been hit based on the logs generated.

Summary

In this chapter we reviewed how to install two very important tools: Portal Toolkit and ILWWCM. We also discussed problem analyses and using Tivoli Web Site Analyzer helped you understand the performance and user characteristics of your Web site. We slightly touched on the Tivoli Web Site Analyzer with the objective that you just understand the power and wealth of information that it can provide.

Now that we have dealt with installation issues in detail, the next logical step is to discuss migration, which is the topic of Chapter 6.

CHAPTER

6

Migrating to WebSphere Portal Version 5.0

It would be great if you didn't have to worry about migration. However, you have dealt with a previous version so you have to deal with it even though it can be a tedious and difficult process. In this chapter, we will discuss the supported migration paths for WebSphere Version 5.0 and show you how to migrate the security attributes, portlets, skins, and themes. Due to the nature of this topic, this coverage is straightforward. For the purpose of being succinct, we will mention our examples using Windows format; however, you can easily determine the Unix/Linux format for 95 percent of the examples by using the appropriate directory format and replacing `WPMigrate` with `WPMigrate.sh`. In this chapter, `<wp4_root>` refers to the WebSphere Portal 4.x root directory, `<wp5_root>` refers to the WebSphere Portal 5.x root directory, `<was4_root>` refers to the WebSphere Administration Server 4.x root directory, and `<was5_root>` refers to the WebSphere Administration 5.x root directory.

Preparing for a Migration

In order to prepare for a migration, you must identify the supported migration path, any constraints, and your environment. There are numerous constraints on the migration paths supported by WebSphere 5.x. Table 6-1 lists the various versions of WebSphere Portal Server that can be migrated to WebSphere Portal 5.x. Basically, you can only migrate from WebSphere Portal Version 4.1.2 and higher. The left column lists the version that can be migrated to the WebSphere Portal version in the right column.

Table 6-1 Supported Migration Paths

WEBSPHERE PORTAL 4.X	VERSION 5.X VERSION
WebSphere Portal Enable Version 4.1.2, 4.1.3a, 4.1.4, 4.1.5, 4.2, or 4.2.1	WebSphere Portal Enable Version 5.x
WebSphere Portal Extend Version 4.1.2, 4.1.3a, 4.1.4, 4.1.5, 4.2, or 4.2.1	WebSphere Portal Extend Version 5.x
WebSphere Portal Experience Version 4.1.2, 4.1.3a, 4.1.4, 4.1.5, 4.2, or 4.2.1	WebSphere Portal Extend Version 5.x
WebSphere Portal Express Version 4.1, 4.1.5, or 4.2.1	WebSphere Portal Express Version 5.x
WebSphere Portal Express Plus Version 4.1, 4.1.5, or 4.2.1	WebSphere Portal Express Plus Version 5.x

The following are some other constraints:

1. Migration between WebSphere Portal environments across operating system (for example, Linux to Windows) is not supported.
2. Migration across database servers (for example, DB2 to SQL server) is not supported.
3. Migration of bookmarks or internal URLs is not supported.
4. Migration of WebSphere V5.x in a cluster is done by migrating data from the WebSphere Portal V4.x cluster main node to a standalone WebSphere Portal V5.x, making the V5.x server a main node, and then creating new clone nodes from it.
5. Content organizer in WebSphere Portal V4.x has been replaced with Document Manager and WPCP has been replaced with WCM. Presently there is no migration path for either new component.
6. The WebSphere Member service must be cataloged locally.

Recommended Migration Environment

To migrate WebSphere Portal V4.x to WebSphere Portal V5.x, both systems should be on separate machines. While it is possible to run both systems on one machine, the amount of resources required and the complexity of the configuration make it an exercise for a masochist.

The migration should be isolated to WebSphere Portal. Any changes in LDAP or database version should be made beforehand and verified with WebSphere Portal V5.x. If you are changing LDAP servers, you should

export the LDAP database information as an LDIF file and re-import it into your new LDAP databases.

Before you perform your migration, you need to do the following:

1. Apply the `42_421_Multiplatform_Cumulative_Fixes.jar` (found on CD 2) to WebSphere Portal server V4.2.1.
2. Add the following line to the `XmlAccessService.properties` file in `<was5_root>\lib\app\config\services\`

   ```
   :com.ibm.wps.command.xml.groupexport.GroupExportEngine=-//IBM//DTD
   GroupExport//EN,
   /com/ibm/wps/command/xml/groupexport/GroupExport.dtd.
   ```

3. If WPS 4.x uses an external security manager, copy the `<wp5_root>\migration\efixes\ESMMigration.jar` file to `<was4_root>\lib\app` directory.
4. Apply fixes `PQ77682_WP50_Migration_iFix` and `PQ77683_WP50_iFix` to the WP V5.x server.
5. Go to your WebSphere Portal 5.x machine and in the directory `<was5_root>\migration\efixes` and copy `WP42X_MP_EXPRESS_Patch.jar` file to `<was4_root>\lib\app` directory on your WPS 4.2.1 machine. If you have a WPS 4.1 machine, copy `WP41X_MP_EXPRESS_Patch.jar` file. Extract the update by typing **jar-xvf WP42X_MP_Express_Patch.jar**, which will result in the file `42X_fix.jar`. Then extract `42X_fix.jar` by typing **jar-xvf 42X_file.jar**.

Remember that before applying fixes, you should stop the servers (Server1 and WebSphere Portal) and restart them after the fixes have been applied. Also verify that both portals are working by accessing them and clicking on various pages and portlets.

Performing the Migration

You will now find out how to do a migration using an example migration environment like the one shown in Figure 6-1. The goal is to migrate WebSphere Portal 4.2.1 using LDAP authentication and store data remotely on DB2 to a WebSphere Portal 5.0 portal with configuration similar to version 4.2.1 except that the database and LDAP versions have been upgraded. The custom place (Place), the custom pages (Page1 to Page3), and the portlets (Portlets1, Portlets2, and Portlets3) will be migrated. Assume that, before the migration, all the servers (`sandbox1.rigorconsultants.com` to `sandbox6.rigorconsultants.com`) have been configured and the

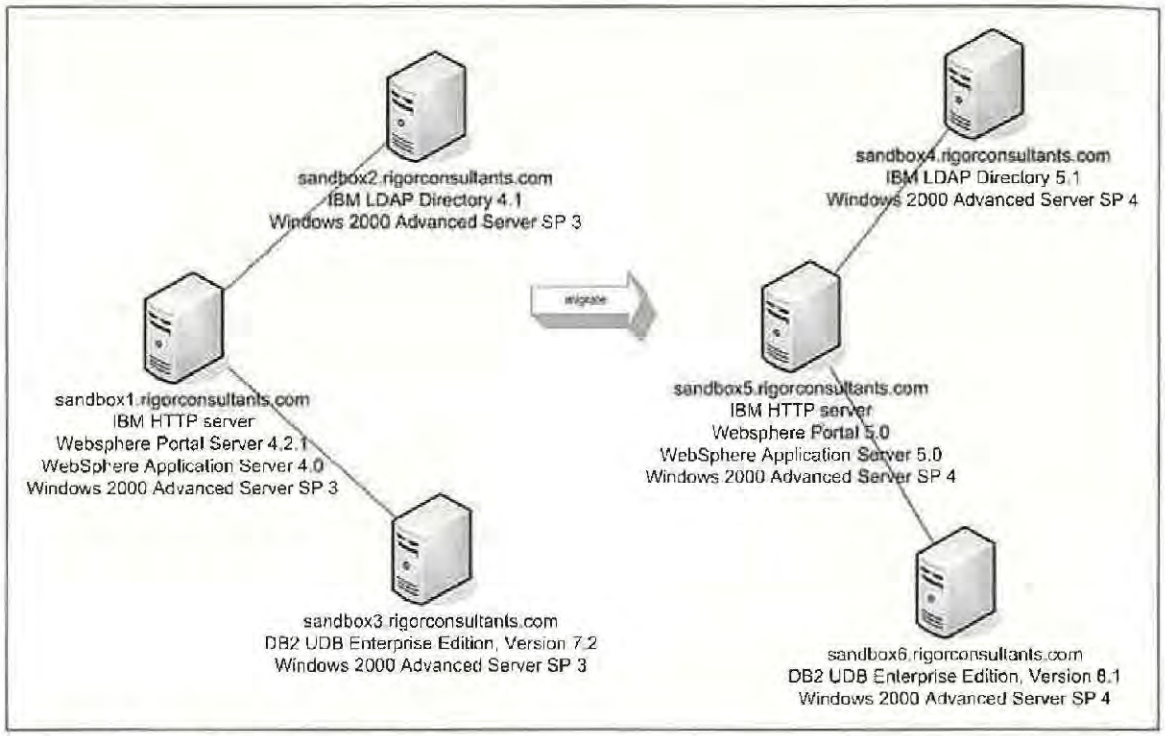

Figure 6-1 Migration example infrastructure.

WebSphere, Web, database, and LDAP software have been tested. Also assume that the appropriate tables and user permissions have been created and set in the remote databases and the user schema and data has exported from `sandbox2.rigorconsultants.com` and imported into the LDAP server on `sandbox4.rigorconsultants.com` using the LDIF utility. All servers and software are running during the migration process. Lastly, assume that both WebSphere Portal versions are configured and get requests through IBM HTTP Server using port 80.

Setting up the Migration Property Values

Much of the migration process is automated using defined automated tasks. To run these tasks, you have to set the values in the migration property files. You will find the migration property files on the WebSphere Portal 5 server (`sandbox5.rigorconsultants.com`) in the directory `<wp5_root>\migration`. But before you modify the files, copy the WebSphere Portal Server `wpsconfig.properties` files on `sandbox1.rigorconsultants.com` to `<wp5_root>\migration` on `sandbox5.rigorconsultants.com`. If you are migrating WebSphere Portal version 4.1.2, then you must add the `WPFamilyNameKey` to the `wps.properties` file. `WPFamilyName` key identifies the WebSphere Portal packaging. For instance, if you installed WebSphere Portal Enable then you would set `WPFamilyName=enable`.

Table 6-2 Core Migration Properties

PROPERTY	VALUE	DESCRIPTION
WpsHostName4x	`sandbox1.rigorconsultants.com`	Fully qualified host name of the WebSphere Portal V4.2.1 machine; `sandbox1.rigorconsultants.com`
WpsPort4x	80	Port number to which WebSphere Portal V4.2.1 receives requests
WpsContextRoot4x	wps	Base URI for WebSphere Portal V4.2.1
WpsDefaultHome4x	portal	Default home for WebSphere Portal V4.2.1
PortalAdminId4x	wpsadmin	WebSphere Portal V4.2.1 administrator ID
PortalAdminPwd4x	wpsadmin	WebSphere Portal V4.2.1 administrator password
wpsinstallLocation4x	`C:/WebSphere/PortalServer`	Root directory of WebSphere Portal V4.x
includePlaces	Place1	Comma-separated names of the custom places you want to migrate
includePages	Page1, Page2, Page3	Comma-separated names of the custom pages you want to migrate
includeApps	Portlet1, Portlet2, Portlet3	Comma-separated names of the portlet applications you want to migrate

The first migration property file to modify is `mig_core.properties`, which specifies the core migration properties. Table 6-2 lists the properties we set for our example.

Migrating Access Controls

Now that your migration environment is all set up, the first step is to convert the access control. This is significant migration since access controls have changed from permission base in WebSphere Portal Version 4 to role-based

Table 6-3 Mapping of Permissions to Roles

V4.X PERMISSIONS	V5.0 ROLES
View	User
Edit Privileged	User
Manage	Manager
Delegate	Security Administrator
View + Edit	Privileged User
View + Manage	Manager
View + Delegate	Security Administrator + User
Edit + Manage	Manager
Edit + Delegate	Security Administrator + Privileged User or Security Administrator + Editor
Manage + Delegate	Administrator
View + Edit + Manage	Manager
View + Edit + Delegate	Security Administrator + Privileged User or Security Administrator + Editor
View + Manage + Delegate	Administrator
View + Edit + Manage + Delegate	Administrator
Create	Not Applicable in WP 5.0

access control (more on this in Chapter 10). Table 6-3 shows the migration process mapping of permissions to roles.

Inheritance blocks will be created by the migration process for all roles except the Administrator and Security Administrator.

To migrate the access controls, do the following tasks:

1. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wp5_root>\migrate` directory.
2. Enter **WPmigrate migrate-user-groups-ac**.
3. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
4. Enter **WPmigrate optimize-roles**. This will eliminate obsolete roles and role blocks.

5. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
6. On WebSphere Portal V5.0 portal, go to Administration ➪ Access ➪ Users and Group Permission and review carefully that all the roles assigned to the Users and Group are correct and set the inheritance blocks as per your company policies.

Migrating Extended User Attributes

In WebSphere Portal 4.x, user attributes are stored in the WebSphere Member Services database while in WebSphere Portal 5.x, user attributes are stored in WebSphere Member Manager. If you just use the seven default user attributes (uid, userPassword, cn, sn, mail, preferredLanguage, and givenName) and do not use a look-aside database (as in 95 percent of the cases), then you can skip this section. If you have extended user attributes, then you must take the following steps:

1. Copy the files `attributeMap.xml` and `attributeMap.dtd` from `<wp4_root>\wms\xml` directory to the `<wp5_root>\template\wmm\xml\migration` directory.
2. In the `<wp5_root>\template\wmm\xml\migration` directory, you will find three files: `wmmDBAttributes.xml`, `wmmLAAttributes`, and `wmmMigrationAttributesInfo.xml`. The file `wmmDBAttributes.xml` is edited only if you have a database-only WMS configuration while the file `wmmDBAttributes.xml` is used if your WMS configuration uses both a database and an LDAP. The `wmmDBAttributes.xml` file specifies the WMM attributes that will be inserted into the WMMDBATR table while the `wmmLAAttributes.xml` file specifies the WMM attributes that will be inserted into the WMMLAATR table. The wmmMigrationAttributesInfo.xml is read by the migration utility to determine where the attribute value data is coming from for each attribute in the `wmmXXAttributes.xml` file. The file `wmmXXAttributes` refers to either the `wmmDBAttributes.xml` or the `wmmLAAttributes` file. In WMS, most attributes are stored in the MBRATTR table and their corresponding values are stored in the MBRATTRVAL table. The exception is that a column name for certain WMS tables can become a WMM attribute and the values associated with the attribute are the WMS column data (now that is confusing!).
3. For each attribute that you are migrating, add the following XML code to the appropriate `wmmXXAttributes` file and save it:

```
<attributeMap wmmAttributeName=subdepartment
              pluginAttributeName=subdepartment
```

```
applicableMemberTypes=Organization
dataType=String
valueLength=256
multiValued=true />
```

where `wmmAttributeName` is the attribute name in WMM and `pluginAttributeName` is the plug-in attribute name in LDAP, `applicableMemberTypes` is the member type which the attribute is associated with, `dataType` is the attribute data type, and `multiValued` is a flag specifying whether the element is multivalued. `ValueLength` is an optional field that applies only to string data type and specifies the maximum length of the attribute value.

4. Now edit the `wmmMigrationAttributesInfo.xml` file and for each attribute (that does not come from a WMS column name) add the following XML code:

```
<AttributeInfo tableName=MBRATTR
                        wmsAttributeName=subdepartment
                        attributeType=String
                        wmmAttributeName=subdepartment />
```

where `tableName` is the table where the attribute is coming from (usually MBATTR), `wmsAttributeName` is the WMS attribute name, `attributeType` is the attribute data type, and `wmmAttributeName` is WMM attribute name. If the attribute comes from a WMS column name then use the following XML code instead:

```
<AttributeInfo tableName=wms_table
               wmsColumnName=wms_column
               attributeType=String
               wmmAttributeName=registration />
```

where `tableName` is the table where the attribute is coming from (usually MBATTR), `wmsColumnName` is the column name of `wms_table`, which contains the attribute value data, `attributeType` is the attribute data type, and `wmmAttributeName` is WMM attribute name. If the attribute comes from a WMS column name then use the following XML code instead;

5. Next you have to edit the `mig_wmm.properties` file found in `<wp5_root>\migration`. This file only needs to be modified if you are using a look-aside database. Table 6-4 shows the values you would need if you were using a look-aside database for our example infrastructure.

6. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wps_root>\migrate` directory.

7. Enter **WPmigrate migrate-wmm**.

Table 6-4 Properties to Migrate Look Aside Database

PROPERTY	VALUE	DESCRIPTION
WmsDbName	wps42db	Alias name of WebSphere Portal V4.x database
WmsDbUrl	jdbc:dbc:wps42db	WebSphere Portal V4.x database JDBC URL
WmsDbUser	db2admin	db2 administrator user ID
WMSDbPassword		db2 administrator password
WmmConfigType	2	LDAP + look-aside database configuration
WmmUid	DB1	The value of the UUID element. The UUID is an ID that uniquely identifies the member from all other members. You can find the value in `<wp5_root>\shared\app\wmm\wmm.xml` file under the node databaseRepository
Db2Home	C:/SQLLIB	DB2 root installation directory on `sandbox6.rigorconsultants.com`
LDAPHasUid	true	States if your LDAP already has a UUID attribute defined
LdapUuidName	ibm-appUUID	LDAP UUID Name

8. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
9. On WebSphere Portal V5.0, check the `trace.log` file under `<wp5_root>\templates\wmm` directory for errors and check the `loader.log` file in the directory `<wp5_root>\template\wmm\db2\database` or `<wp5_root>\template\wmm\db2\lookaside` to see if all the rows are loaded correctly.

Migrating Themes, Skins, and Style Sheets

The migration of themes, skins, and style sheets that you may have customized is a manual process. The steps to do it using our example infrastructure are as follows:

1. Copy the WPS 4.2.1 customized themes and skins directories to `<was5_root>\installedApps\sandbox5\wps.ear\wps.war` on the WP V5.0 server.

2. Scan the JSPs used by the themes and skins and change the following:
   a. Remove the following tags: `<wps:pageGroupLoop/>`, `<wps:pageGroup/>`, `<wps:pageLoopInit/>`, `<wps:page/>`, and `<wps:frameURL/>`. They are no longer supported.
   b. Replace `<wps:pageLoop/>` by `<wps:navigationLoop/>`, `<wps:pageLoopShift>` by `<wps:navigationShift/>`, `<wps:pageRender/>` by `<wps:compositionRender/>`, and `<wps:URLSelect>` by the `<URLGeneration/>` tags.
   c. The behavior of `<wps:text/>` and `<wps:textParam/>` has changed. If the content is provided in the body of the `<wps:text/>` tag, then it will always be written to the page.
   d. The attributes `computeSelectionPath` and `useModelFromRequestAttributeNamed` of the `<wps:navigation/>` tag are no longer supported.
   e. For `<wps:if/>` and `<wps:unless/>` tags, the attributes `error`, `pageGroupSelected pageSelected`, and `frameMode` are no longer supported. The attribute `pageGroupAvailable` has been replaced by `nodeInSelectionPath` and `pageAvailable` has been replaced by `navigationAvailable`.
3. Update the style sheets used by the themes by looking at the Science theme in WP V5.0, find the classes that have a comment associated with them, New in V5, and copy them to your theme. Then modify the classes to match the look of your theme.
4. Log in to `http://sandbox5.rigorconsultants.com/wps/portal` and go to Administration ➪ Manage Themes and Skins.
5. Add your customized skins.
6. Test the themes and skins by creating a root page using the Page Customizer, assign the customized theme to the new page, and assign some portlets.
7. Go to the new page and test to see if the new page is similar to the customized skin and themes in your WPS 4.2.1 portal.

Migrating Portlet Applications

The next step is to migrate your portlet applications. Copy your custom portlets from `<wp4_root>\deployed` on WPS 4.2.1 (`sandbox1.rigorconsultants.com`) to a temp directory on your WP 5.0 machine (`sandbox5.rigorconsultants.com`). Expand the WAR file. If your portlets worked on WP 4.2.1 then minimal modifications are required of the source code. The only changes you might need are the following:

1. Methods `getAttribute()`, `getAttributeNames()`, and `getLastLoginTime()` are no longer supported.
2. The portlet helper classes JSP, File server, and RSS (XSLT) portlet have been deprecated.
3. `PortletURI.addAction(java.lang.String simpleaction)` should be used instead of `PortletURI.addAction(PortletAction action)`.
4. `ActionEvent.getActionString ()` should be used instead of `ActionEvent.getAction`.
5. `PortletRequest.getUser()` must be used instead of `PortletSession.getUser()` especially since `PortletSession.getUser()` returns a null value.
6. Make sure there is no `portlet.tld` file in the WAR file. If there is, remove it.
7. If you used Java Authentication and Authorization Service to access the user credentials, you must rewrite the code using the credential vault service.
8. Click-to-action portlets behave differently in WP 5.0. Output parameters will not automatically propagate unless they are explicitly wired to input parameters. You may have to rewrite them as a cooperative portlet, which is just an enhance click-to-action portlet.
9. You need to copy the `pbportlet.jar` file found in `<wp4-root>\pb\lib` directory on you WPS 4.2.1 server `(sandbox1.rigorconsultants.com)` to each of your migrating click-to-action portlets `WEB-INF\lib` directories. If within the click-to-action portlet WAR file there is another copy of the `pbportlet.jar` file in another location, then you must remove it to avoid classloading conflicts.
10. Ensure you are not referencing any deprecated tags mentioned in the previous sections.
11. Make sure that multiple portlets within a portlet application do not point to the same servlets definition in `web.xml`.

After you have modified the files, archive the files back into a WAR file and copy them to `<wp5_root>\installableApps` on your WP 5.0 server. Next perform the following tasks:

1. Ensure that in the `mig_core.properties` file, you have set the includeApps property with the names of all your custom portlets.

2. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wp5_root>\migrate` directory.
3. Enter **WPmigrate export-apps**.
4. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
5. On WebSphere Portal V5.0 portal, verify that all your portlet applications that you migrated are present under Portlets ⇨ Manage Applications ⇨ Web Modules. Click Modify Parameters and see that they are correct. Next go to Administration ⇨ Access Resource Permissions and under Resource Permissions click Portlet Applications and then verify that the access control settings are correct. Lastly, add them to a test page and perform a function test.

Migrating Places

In WebSphere Portal V5.0, the concept of a Place was eliminated. The migration tool converts them to a top-level page. Do the following tasks to migrate them:

1. Ensure that in the `mig_core.properties` file, you have set the includePlaces property with the names of all your custom Places.
2. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wp5_root>\migrate` directory.
3. Enter **WPmigrate migrate-places -DdeployPages=false-DdeployApps=false -DconfigThemesSkins=false**.
4. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
5. On WebSphere Portal V5.0 portal, click on Administration ⇨ Portal User Interface ⇨ Manage Pages. Then under Content Root, click on My Portal and verify that your Places were migrated. Check the page properties, including the properties under Advance Options to see if they are correct.

Migrating Pages

1. Ensure that in the `mig_core.properties` file, you have set the includePages property with the names of all your custom Pages.
2. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wp5_root>\migrate` directory.

3. Enter **WPmigrate migrate-pages -DconfigThemesSkins=false**.
4. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
5. On WebSphere Portal V5.0 portal, click on Administration ⇨ Portal User Interface ⇨ Manage Pages. Then under Content Root, click on My Portal and verify that your pages were migrated. Check the page properties, including the properties under Advance options to see if they are correct.

Migrating All User Customizations

This task migrates all skins referenced by themes.

1. Ensure that in the `mig_core.properties` file, you have set the includePages property with the names of all your custom Pages.
2. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wp5_root>\migrate` directory.
3. Enter **WPmigrate migrate-user-customization**.
4. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
5. On WebSphere Portal V5.0 portal, click on Administration ⇨ Portal User Interface ⇨ Manage Pages. Then under Content Root, click on My Portal and verify that all your customization were migrated.

Migrating Credential Vault Slots and Segments

1. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wps_root>\migrate` directory.
2. Enter **WPmigrate migratecredential-slots-segments**.
3. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
4. On WebSphere Portal V5.0 portal, click on Administration ⇨ Credential Vault ⇨ Manage Vault Segments and verify that the segments have been migrated. Then click Administration ⇨ Credential Vault ⇨ Manage Vault Slots and verify that your vault slots were migrated.

Migrating Credential Vault data

After you have migrated the credential vault slots and segments, you have to migrate the actual data. You do this by exporting the data from the tables

VAULT_DATA and VAULT_RESOURCES found on your WPS V4.0 database (`sandbox3.rigorconsultants.com`) and importing them to your WP V5.0 database on `sandbox6.rigorconsultants.com`. The tables on the VP V5.0 database gave the same names: VAULT_DATA and VAULT_ RESOURCES.

Verify the migration by clicking Access ⇨ Credential Vault ⇨ Manage System Vault Slots and reviewing the data for accuracy and completeness.

Migrating Everything at One Time

If you are feeling very confident (and lucky), you can do a complete migration with one command by following the steps given below:

1. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wp5_root>\migrate` directory.
2. Enter **WPmigrate migrate-all**.
3. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.
4. On WebSphere Portal V5.0 portal, verify the migration by reviewing all your pages, portlets, credential vault slots/segments, and all the access controls for your users, groups, and resources.

XML Configuration Interface

Another tool you can use for migration and batch changes is the XML Configuration Interface. This tool enables you to export a part of or entire configuration and then re-create it on another portal. You can also use XML Configuration tool to help back up or restore complete portal configuration, copy parts of a configurations, install additional resources, or perform reproducible administration tasks.

The XML Configuration can be used to perform a complete re-creation of a portal. It cannot deal with War files, cannot be used for WPCP, and is not suitable for large volumes of data. Backup and restoration should be performed by regular database and file backup tools.

XML Config is invoked by typing **xmlaccess** on the command line. It can be invoked on the actual portal server or on any client. To invoke it on a remote client you need to have a Java engine and the `tool.jar` file that is found in `<wp_root>/bin` directory.

As can be seen in Figure 6-2, the XML Config tool takes an input file consisting of XML tags (1), sends it by HTTP from the client (2) to the WP

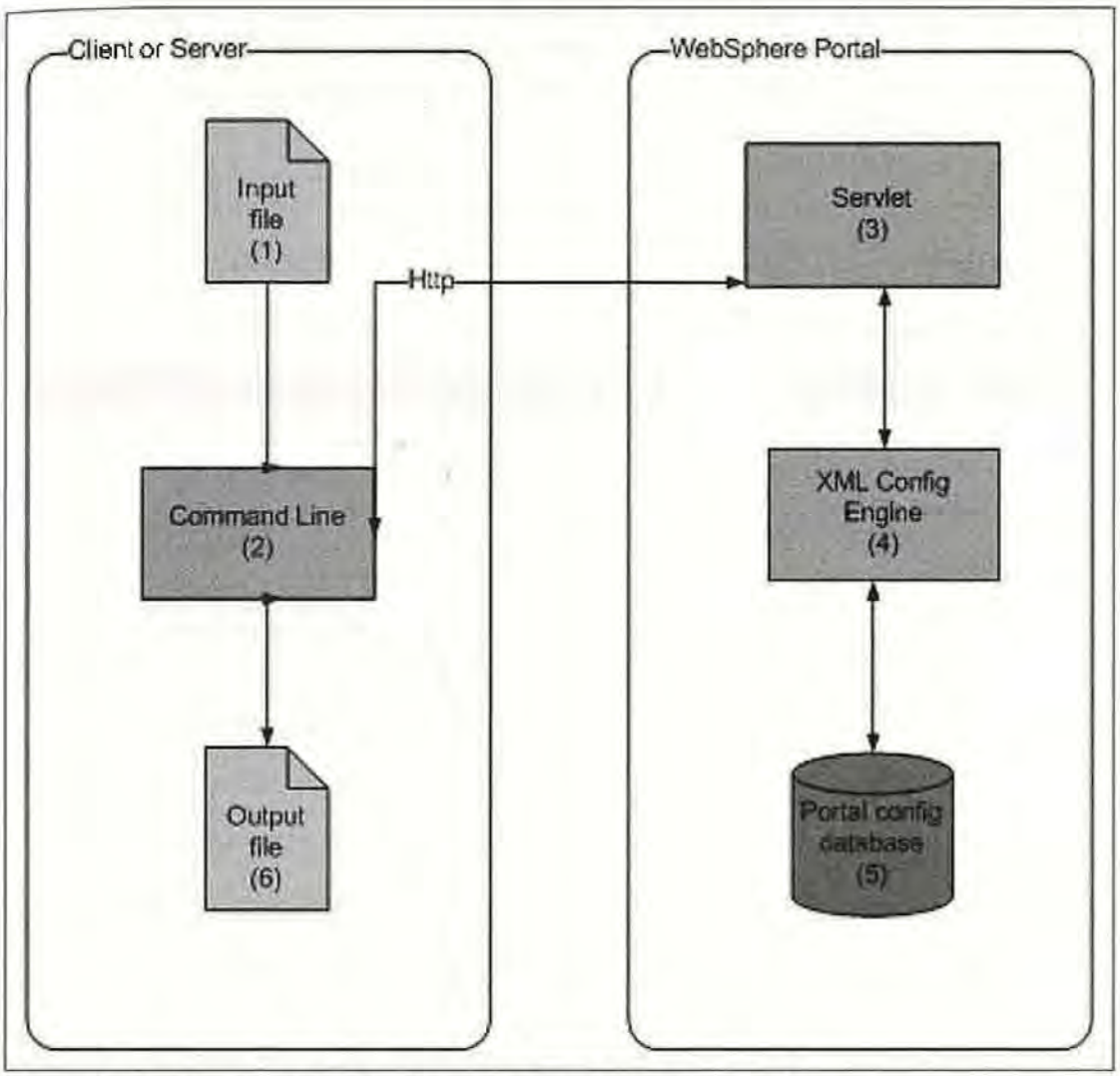

Figure 6-2 How the XML Configuration Interface utility works.

servlet (3), which passes the data to the XML Config engine (4). The XML Config engine stores and retrieves data specific to the configuration actions from the Portal config database (5). The output is sent back to the client via HTTP as XML tags (6), which can then be used as input.

The XML Configuration Interface handles two types of requests: Export and Update. Export results in complete or partial portal configuration into XML. Update causes the portal configuration to be modified based on the XML script values.

The syntax for the command is as follows:

```
xmlaccess-in inputfile-user user-pwd password-url URL-out
output_file
```

where `-in` is the input file, `-user` is the authorized portal user id, `-password` is the password, `-url` is the portal URL for processing the command (such as `http://www.rigorconsultants.com/wps/config`), and `-out` is the output file.

The user must be a super user with manager role on XML_ACCESS and administrator role on portal. The usernames and passwords are sent unencrypted, so send them over the intranet or use VPN.

The XML Configuration interface models your portal as XML tags with the defined hierarchical structure shown in Table 6-5. (This table can also be found in the Portal infocenter.)

Table 6-5 Description of XML Access Tags

XML	DESCRIPTION
portal	Main element of every XML request
global-settings	Global portal settings
services-settings	Global portal settings for portal services
virtual-resource	Virtual resources that have associated access control settings
user	Users defined in the portal user management system
group	Groups defined in the portal user management system
markup	Markups that can be supported by portal pages
client	Client devices (browsers) that the portal knows about
event-handler	Definitions of event handlers that can react to events in the portal
skin	Visual appearance settings that can be applied to user interface elements
theme	General visual settings that can be applied to the user interface
web-app	Web modules containing portlets
servlet	Servlets that are defined in the Web module
portlet-app	Portlet applications that are defined in the Web module
portlet	Portlets that are defined in the portlet application
content-node	Elements of the portal content tree (pages or labels)
component	Layout components of pages
portletinstance	Occurrences of a portlet on a page with customized settings
credential-segment	Segments for storing credentials in the credential vault
credential-slot	Slots in a credential segment that hold a credential
url-mapping-context	User defined URLs that map to pages in the portal

Obviously, the XML files generated can be quite complex and cryptic, especially with a detailed portal.

XML Input File

The XML Configuration Interface requests are defined in an input file with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
    <request
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="PortalConfig_1.2.xsd"
    type="export | update">
      <portal . . . >
      definition of configuration parts to be exported or updated
      </portal>
      <status . . . >
      success or failure indication for the processing
      </status>
   </request>
```

You must always use the schema reference with no namespace to the schema "`PortalConfig_1.2.xsd`."

Actions are carried based on the value of the request type parameter. You have two types of requests: update or export. Update performs a modification to the portal while export is used for extracting information.

Each request has actions associated with it. For update, only locate, create, update, and delete are permitted while export is limited to locate and export.

The locate action finds the portal resource associated with a specific XML element. Create action creates a portal resource with attributes you specify. Update action modifies the corresponding portal resource with a given configuration data. Delete action removes the portal resource based on the specified XML element. Finally, export action generates an XML representation of the portal resource.

To understand how to set up a configuration request, let's walkthrough a sample request file provided by IBM. You can find this code at `<WebSphere>\PortalServer\doc\xml-samples`. Normally you would look for a sample that performs a function similar to what you want to do and perform any necessary modifications. This XML request file will create the user Homer Simpson and add it to the group Springfielders. Note that in the update request, we set create-oids to "true." This enables object IDs to be symbolically created and valid only within the XML script which prevents the portal from choosing an already existing object-id and overwriting it.

```
<?xml version="1.0" encoding="UTF-8"?>
<request
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
        xsi:noNamespaceSchemaLocation="PortalConfig_1.2.xsd"
        type="update" create-oids="true">
                <portal action="locate">
                <user action="update" name="hsimpson" firstname="Homer"
                lastname="Simpson" password="secret">
                        <description>My favorite TV star</description>
                        <parameter name="preferredLanguage" type="string"
                                update="set">en</parameter>
                                </user>
                <group action="update" name="springfielders">
                        <description>Inhabitants of Spring
                        field</description>
                        <member-user update="set" id="hsimpson"/>
                </group>
                </portal>
</request>
```

XML Configuration Interface Special Properties

The XML Configuration Interface has some special properties of which you should be aware. Since the portal is represented in XML, the XML Configuration utility must have some way of representing the order of elements on a GUI screen or the sorting order. This is done using three methods. If the resource is identified as an integer, it will be sorted into position relative to the ordinals of existing siblings. If a position indicator is specified (hash (#) mark followed by an integer), it will appear in that position in its content parent. Lastly, you can specify the special values with first and last, which indicates that the resource is sorted either in the first or last position.

You should also be careful with unique names in XML scripts especially if these scripts run on many portal installations. If a unique name is assigned by the XML script and that unique name is already used on the system, the script will fail.

XML Configuration Interface Transactional Support

The XML Configuration Interface provides limited transactional support. Changes in the portal database are grouped into transactions when you create, update, or delete resources. All changes that are part of the transaction are either executed completely or not executed at all.

The main request element has a transaction attribute, which defines the grouping value. The value can be Resource (default value), which means that every top-level resource is processed in one separate transaction or Request, which means the entire XML script is executed in one transaction.

The transactional support is limited because it only applies to changes to the portal database. This means changes to portlets, users, and groups stored in LDAP, or role assignments in external access control systems cannot be rolled back if a transaction fails. Similarly, if the latter changes failed, changes to the database will not be rolled back.

Changes in XML Configuration Interface for Version 5.0

The XML Configuration Interface utility is not compatible between different versions of WP. Version 5.0 of the utility has been enhanced using the following steps:

- Use of XML Schema instead of DTD file.
- Support for mandatory type attribute.
- Support for event handlers, URL Mappings, and property broker wires. A wire connects a source and a target portlet instance enabling changes to the source to be propagated to the target.
- Ability to manage users and groups.
- Support of object IDs, unique names, and the ordinal attribute
- Support of role administration.

In version 4.2 the portal navigation structure was expressed as a tree of components of type layered inside Root.Composition while in WebSphere Portal version 5.0, labels and pages are organized in a tree that is expressed by their content-parentref attributes. The navigation structure in the portal is the same as the structure of the content tree. There is no root composition defined in the XML scripts anymore.

Do the following to migrate your XML Configuration scripts to WP V5.0,

1. Open a command prompt on your WebSphere Portal V5.0 machine (`sandbox5.rigorconsultants.com`) and go to `<wps_root>\migrate` directory.
2. Enter

   ```
   WPmigrate migrate-xmlaccess-script -DScriptPath=<path>
   -DScriptSrc=<scriptname> -DScriptDest=<convertedscriptname>
   ```

   where `<path>` is the complete path to your script file, `<scriptname>` is the script file name to be converted, and `<convertedscriptname>` is the name of the converted script file.
3. Wait for the Build Successful message and then check the `MigrationReport.xml` file for any errors.

Summary

In this chapter we reviewed the tasks and tools that are provided to assist you in migration. We clearly did not cover all the situations such as migration of collaboration portlets, portlets based on struts, portlet toolkit, and portlets using transcoding technology. We also did not bother with migration of WPCP and Personalization since they are both being replaced by WCM.

The purpose of this chapter was to help you in the migration of the most common components. For more detail information, we refer you to the WebSphere Portal Migration manual.

At the end of the chapter, you are probably saying to yourself, "boy this is complicated." Well it is, and partly this is due to the rapid infusion of functionality with each version of WebSphere Portal. With version 5.x, you are seeing the product mature and the product becoming simpler to use. We believe that migrations in the future will become easier as more effort is put into delivering configuration and migration wizards.

Well, now that you have got WebSphere Portal up and running, it is now time to actually use the product. In the next chapter, you will learn how to easily define the look-and-feel components of your portal.

PART II

Building and Administering Portals with WPS

CHAPTER

7

Defining Portals and Pages

Portals and pages provide the framework for content delivery within WebSphere Portal. Everything the user sees upon entering your portal is part of a page. In this chapter you will learn the basic skills required to define and customize the content delivered by your portal.

Pages are essentially scaffold around which your portal is built, and in this chapter we focus on defining portal structure. The concepts involved are quite simple, and we will guide you through the most common tasks associated with the construction and customization of your portal.

Portals and Pages

Complex data is usually stored in hierarchical structures, a common example being files and folders in an operating system. WebSphere Portal follows this approach and uses a hierarchical structure to maintain the set of resources and components used to generate the user experience.

In order to view the structure of the portal, you click the Administration tab in the portal toolbar. To do this, you must be logged in as an administrator. As we saw in Chapter 2, to log in as administrator, you need to access the URL `http://<hostname>:<port>/wps/portal`, where *hostname* is the server on which WebSphere Portal Server is installed and port is 9081 (unless you modified the default settings), and click Log in. Once you are in the Administration tab, click Manage Pages under Portal User Interface. WebSphere Portal Server will configure your Portal Settings.