IBM WebSphere Portal
V4.1 Handbook
Volume 3

Understand the IBM WebSphere Portal architecture

Step-by-step installation instructions for IBM WebSphere Portal

Implement new and enhanced capabilities of IBM WebSphere Portal

Rufus Credle
Denise Hendriks Hatzidakis
Sunil Hiranniah
Gord Niguma
Dwight Norwood
Roshan Rao
Bernhard Stimpfle

ibm.com/redbooks

This edition applies to IBM WebSphere Application Server Advanced Edition V4.0.2, IBM Secureway Directory V3.2.2, IBM WebSphere Personalization V4.0, DB2 Universal Database V7.2, IBM WebSphere Studio Application Developer V4.02, and IBM WebSphere Portal for Multiplatform V4.1.2.

© Copyright International Business Machines Corporation 2003. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# Contents

**Notices** .......................................................... vii  
**Trademarks** ...................................................... viii  

**Preface** ............................................................ ix  
The team that wrote this redbook .................................................... x  
Become a published author .................................................................... xii  
Comments welcome .............................................................................. xiii  

**Chapter 1. Web content management** .............................. 1  
1.1 Introduction ............................................................... 2  
1.2 Web content management fundamentals ............................ 2  
1.3 Installation ................................................................. 9  
1.3.1 Patched rt.jar file ....................................................... 9  
1.3.2 Remove Lotus Notes clients .............................................. 9  
1.3.3 Install DB2, IBM HTTP Server and WebSphere Application Server .................................................. 10  
1.3.4 Generating keys in WebSphere Application Server ................ 12  
1.3.5 Install Domino components and Web Content Publisher .......... 15  
1.3.6 Configure Domino Administration client .............................. 32  
1.3.7 Configure a workflow for Web Content Publisher .................. 40  
1.3.8 Configuring WebSphere Application Server security ................ 52  
1.3.9 Verify the Web Content Publisher install ............................. 56  
1.3.10 Configure Domino for WebSphere Portal ........................... 61  
1.3.11 Install WebSphere Portal ............................................... 76  
1.3.12 Verify the WebSphere Portal install .................................... 87  
1.3.13 Updating security to enable single sign-on .......................... 87  
1.3.14 Additional configuration for Web Content Publisher .............. 93  
1.3.15 Post-installation ......................................................... 94  
1.4 Web Content Publisher implementation ............................ 99  
1.4.1 Creating users ........................................................... 100  
1.4.2 Creating groups for Lotus Workflow ................................... 105  
1.4.3 Managing Lotus Workflow .............................................. 107  
1.4.4 Creating Web Content Publisher project ............................. 108  
1.4.5 Creating structured content ............................................ 130  
1.4.6 Creating a publishing server .......................................... 141  
1.4.7 Managing versions and editions ....................................... 146  

**Chapter 2. Collaboration** .............................................. 147  
2.1 An overview ............................................................. 148  
2.1.1 Collaborative Components ............................................ 148
5.3.1 Portal reports .................................................. 226
5.3.2 Benefits .................................................... 226
5.4 Planning ....................................................... 227
  5.4.1 Supported platforms ........................................ 227
  5.4.2 Prerequisites .............................................. 227
  5.4.3 Disk space considerations .................................. 227
  5.4.4 Database considerations ................................... 228
  5.4.5 Application Server considerations .......................... 228
  5.4.6 Remote file system considerations ......................... 229
5.5 Installation using Portal Setup Manager ...................... 229
  5.5.1 Creating the Site Analyzer administrative database .... 230
  5.5.2 Installing Site Analyzer .................................... 231
5.6 Using Site Analyzer ........................................... 240
  5.6.1 Configuring NCSA Combined logging for IBM HTTP Server .... 241
  5.6.2 Configuring logging for WebSphere Personalization ........ 241
  5.6.3 Configuring logging for WebSphere Portal .................. 242
  5.6.4 Creating a Site Analyzer project ............................ 244
  5.6.5 Importing log files into Site Analyzer ...................... 249
  5.6.6 Creating a sample Portal report ............................ 258

Abbreviations and acronyms ..................................... 265

Related publications ............................................. 267
IBM Redbooks ................................................... 267
Other publications ............................................... 267
Online resources .................................................. 268
How to get IBM Redbooks ........................................ 269

Index ............................................................. 271
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.
Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- AIX®
- DB2®
- DB2 Universal Database™
- Domino™
- @server™
- IBM®
- iNotes™
- Lotus Discovery Server™
- Lotus Notes®
- Lotus Workflow™
- Lotus®
- Notes®
- QBIC®
- QuickPlace™
- Redbooks™
- Redbooks (logo)™
- Sametime®
- SecureWay®
- Tivoli®
- VisualAge®
- WebSphere®
- xSeries™

The following terms are trademarks of other companies:

- ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

- Microsoft, Windows, Windows NT, Windows 2000 and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

- Red Hat, the Red Hat "Shadow Man" logo, RPM, Maximum RPM, the RPM logo, Linux Library, PowerTools, Linux Undercover, RHmember, RHmember More, Rough Cuts, Rawhide and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

- Linux is a registered trademark of Linus Torvalds.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

- C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

- UNIX is a registered trademark of The Open Group in the United States and other countries.

- SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

- Other company, product, and service names may be trademarks or service marks of others.
Preface

The *IBM WebSphere Portal V4.1 Handbook* is available in three volumes of Redbooks. This is Volume 3.

These IBM Redbooks position the IBM WebSphere Portal for Multiplatforms as a solution that provides a single point of interaction with dynamic information, applications, processes, and people to help build business-to-employee (B2E), business-to-business (B2B), and business-to-consumer (B2C) portals.

WebSphere Portal consists of three packaged offerings:
- Portal Enable
- Portal Extend
- Portal Experience

In the three volumes of the *IBM WebSphere Portal V4.1 Handbook*, we cover WebSphere Portal Enable and Extend.

The *IBM WebSphere Portal V4.1 Handbook* will help you to understand the WebSphere Portal architecture, teaches how to install and configure WebSphere Portal and how to administer portal pages using WebSphere Portal, discusses the development of WebSphere Portal portlets, and covers how to use specific WebSphere Portal applications.

Across the volumes of the IBM WebSphere Portal, you will find step-by-step examples and scenarios showing ways to rapidly integrate your Enterprise Applications into an IBM WebSphere Portal Server environment using state-of-the-art technologies, such as portlets, and implementing new and enhanced capabilities incorporated in the current releases of IBM WebSphere Portal Server offerings, such as access controls and page customization using themes and skins.

In this redbook, we discuss the WebSphere Portal applications and their uses.

A basic knowledge of Java technologies such as servlets, JavaBeans, EJBs, JavaServer Pages (JSPs), as well as XML applications and the terminology used in Web publishing, is assumed.
The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Rufus Credle is a Senior I/T Specialist and certified Professional Server Specialist at the International Technical Support Organization, Raleigh Center. He conducts residencies and develops redbooks about network operating systems, ERP solutions, voice technology, high availability and clustering solutions, Web application servers, pervasive computing, and IBM and OEM e-business applications, all running on xSeries systems. Rufus's various positions during his IBM career have included assignments in administration and asset management, systems engineering, sales and marketing, and IT services. He holds a BS degree in business.
management from Saint Augustine’s College. Rufus has been employed at IBM for 22 years.

**Denise Hendriks Hatzidakis** is a managing director and WebSphere Architect with Perficient, Inc. Denise has a BS in Physics and a BS degree in Computer Science, followed by a MS in Electrical and Computer Engineering. She joined IBM and spent 10 years as a lead developer for VisualAge and WebSphere in various capacities. She has recently joined Perficient, Inc., where she makes extensive use of her skills as a consultant in WebSphere and J2EE technologies.

**Sunil Hiranniah** is a Software Engineer and works for IBM Developer Relations Technical Support Center in Dallas, USA. He has over five years of experience in the software industry working for various commercial projects. He has wide experience with WebSphere Portal, WebSphere Application Server, J2EE and databases, and has written and published extensively on the WebSphere family of products.

**Gord Niguma** is an IT Specialist for the Vancouver Innovation Centre in IBM Canada. He has six years of experience in the Web development field, working for customers such as Air Canada and the NHL Players Association. He holds a Masters degree in Computer Science from Simon Fraser University and a Bachelor of Science in Computer Science from Dalhousie University. His areas of expertise include portals and Web content management.

**Dwight Norwood** is a Director and Senior Consultant for Courtbridge Consulting Group, an IBM Business Partner located in East Granby, Connecticut (U.S.A.). He has 30 years of experience in information technology, with 10 years of Lotus Notes and Domino experience. A graduate of the University of Notre Dame, he holds a Master's degree in Computer Science from Rensselaer Polytechnic Institute and a Master's degree in Business Administration from the University of Connecticut. He has written extensively on Notes and Domino development. He has special interests in enterprise knowledge management and publishing, and Web-related security.

**Roshan Rao** is a Senior Consultant with Perficient Inc., with three years of experience in design and development of object-oriented systems. He has a degree in Commerce from the University of Mumbai and is currently pursuing a Masters degree in Computer Science from Maharishi University of Management. He is an IBM Certified Specialist for WebSphere Application Server and WebSphere MQ. His key area of work includes Java technologies, portals, messaging and Enterprise Application development and integration.

**Bernhard Stimpfle** is a Pervasive Solutions Architect for the IBM Pervasive Computing Division in Boeblingen, Germany. He reviews architectures, implements customer-specific product add-ons and supports major customers on site in critical situations. He has spent eight years in the IT industry, working for
Daimler-Chrysler Aerospace and managing his own business. His area of expertise include pervasive computing, UNIX, Java 2 Enterprise Edition (J2EE) programming, and solution architectures. He is a Red Hat Certified Engineer (RHCE) and holds a Diplom-Ingenieur degree in Computer Science from Berufsakademie Ravensburg, Germany.

Thanks to the following people for their contributions to this project:

Gail Christensen, Cecilia Bardy, Margaret Ticknor, Tamikia Barrow, Diane O’Shea
IBM International Technical Support Organization, Raleigh Center

Mark C Fullerton, Consulting I/T Architect
IBM Ontario

Vishy Gadepalli, Stacy Joines and Sung-Ik So
IBM WebSphere Enablement and Consulting Team, Raleigh

Axel Buecker, ITSO Project Leader
IBM Austin

Stefan Schmitt, Marian Puhl, Ingo Schuster, David S. Faller
IBM WebSphere Portal Development, IBM Boeblingen

Theodore Buckner
IBM Pervasive Computing Division, Raleigh

Frank Seliger
IBM Pervasive Computing Division, Boeblingen

Tim Orlowski
IBM WebSphere Beagle Validation Team Lead, Raleigh

**Become a published author**

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.
Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks

- Send your comments in an Internet note to:
  redbook@us.ibm.com

- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HQ7 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195
Web content management

This chapter covers creating, approving, and publishing Web content. It describes features and functions only as they relate to system administrators. It is not intended as a full “how-to” guide for developers and administrators of the Web Content Publisher application.
1.1 Introduction

Web Content Publisher is a Web content management system that allows non-technical users to publish content to the Web site using simple Web forms. It supports a multi-user environment by managing workflow, security, administration and editioning.

This section is written from a system administrator’s perspective. It is not designed to describe the features and functions of Web Content Publisher.

Tip: For a “how-to” guide to using the Web Content Publisher, see the help files. The files are stored at http://<yourhost>/wps/wcp/helpsystem/en/docFrameset.html by default and are available after the installation of Web Content Publisher. An excellent tutorial is available by clicking the Getting Started tab then clicking Tutorial in the left-hand navigation bar.

1.2 Web content management fundamentals

Web content management provides an environment for users to create, manage, and publish a Web site. It manages the life cycle of content from a request to create content and the creation of content, to publishing the content.

This section describes the basics of a generic Web content management. It is important that you understand these fundamentals before proceeding with Web Content Publisher specific implementation details.

The following sections describe a scenario of the management of a news Web site. It highlights key aspects of Web content management systems.

Scenario: San Francisco Newspaper

Joe SportsEditor needs a new article on Barry Bonds as he approaches baseball’s home run record. He asks his top San Francisco sports writer, Greg ContentContributor, to put together an article by Thursday.

Greg ContentContributor receives a notification from Joe SportsEditor. Greg needs to publish the article on the Internet but is not familiar with HTML or JSP. He only knows how to write sports articles. Rather than try to write an HTML page himself, he fills out a standard form for headline news articles. The fields he has to fill out include a headline, subject, keywords, author and content body. This form is known as an authoring template. Greg saves his work as an instance of structured content and previews it through a preview template. Everything looks great, Greg is happy with his article, and submits it. He forgot to
enter any keywords, so the article is immediately rejected by the system. The system validates the data and the error is caught before it is sent to Greg’s editor. Greg fixes his mistake and submits it to Joe SportsEditor.

Joe SportsEditor reads the article and decides it needs more work. He rejects it and Greg ContentContributor is notified through an e-mail message. Greg reopens his article through an authoring template devised for editing pre-existing content. Greg revises and re-submits the article to Joe Editor. Joe is happy with the revised article. This approval cycle is part of the Web site’s workflow process.

Joe must convert Greg’s article and add the appropriate look and feel to catch the audience’s attention. Joe knows nothing about formatting, graphics, or HTML, but he has several generation templates that he can choose from. The generation template will convert Greg’s input from the authoring template and add the Web site’s banner on the top, a banner at the bottom, and a navigation pane on the side. The result will be an HTML page containing Greg’s article that has the Web site’s standard navigation and look and feel.

Joe is ready to publish the article. But instead of publishing directly to the production Web site, he publishes to a staging server. Joe’s project only covers the sports section. The staging site’s administrator is Tara WebMaster. She verifies all submissions on the Web site, including other projects such as World News and Entertainment.

At midnight, Tara makes an edition of the Web site. This edition represents a snapshot of all approved articles. Once the edition is created, Tara publishes it to the production server. She schedules publishing to begin at 3 a.m.

This sample scenario illustrates the life cycle of Web content. It illustrates the key features of Web content management systems. We will examine each of these areas.

**Authoring templates**

Authoring templates are used for creating, editing, and viewing content. In this scenario, Greg ContentContributor used the common template to input his sports article into the Web content management system. Once the data is input into the authoring template and stored, the generation template aggregates the data with a “look-and-feel”, including managing banner graphics and page navigation.

Separating authoring templates from generation templates provides several advantages over simply creating an HTML page:

- Modifying the generation template does not require a change to the data. For example, if each product had a separate HTML page that was created by a content contributor, changing the banner of the page would require modifying
each HTML page. By separating the data from the presentation, a developer could simply modify the generation template to include the new banner and re-aggregate the data to the new generation template. A content contributor would not even need to modify their data.

- Content contributors do not need to worry about the look and feel of the page and the complexity of HTML and JSP. Developers can focus on creating the generation template, and the content contributor can produce content by filling out a form.

- Supporting multiple format types such as HTML and WML is simplified. Rather than creating a separate HTML and WML file and re-entering the data into each file, the data is entered through an authoring template only once, and processed with two generation templates, one for HTML and one for WML.

- Data validation can be performed when content is input in the system. This ensures that data types, formats, field lengths, etc. are consistent.

The example shown in Figure 1-1 on page 5 shows an authoring template for adding a toy to an inventory. Fields such as Product Number, Name and Description are entered. This ensures that all toys added to an inventory have the same fields, with the proper field lengths and field types.
Figure 1-1 Authoring template

The authoring template may be implemented to handle data validation to ensure consistency of input. For example, in Figure 1-1 the system can verify that the Product Number is unique before allowing the new toy to enter the system.

When an author fills in an authoring template and saves the work, it creates an instance of structured content. For example, in Figure 1-1 the user creates an instance of a toy. The instance is structured data and is usually stored in a relational database or in a structured file format such as XML.

Authoring templates may either be designed for new content or for editing existing content. In the scenario, Greg ContentContributor re-opens his article after it is rejected by Joe SportsEditor. He is using the authoring template for editing existing content.
Preview templates

Preview templates are used for quickly viewing a single instance of structured content that was created by an authoring template. This is done before a generation template has applied all the appropriate formatting that is required before publishing to the Web site.

In the scenario, Greg ContentContributor previews his content before submitting to Joe SportsEditor. This preview of his content is provided through the preview template.

Generation templates

Generation templates are used to generate a view of structured content and store it in a file. The generation template converts the structured content into a format that is publishable to a Web site. The output will be a file such as HTML, WML, or JSP.

When a template is used at runtime to dynamically generate a view of content, it is sometimes referred to as a presentation template. By contrast, generation templates are typically used at development time to produce files that are later published to a Web site. Often, generation and presentation templates can be used interchangeably.

Multiple generation templates may be applied on the same structured data to allow the data to be presented in different file formats. This is particularly useful to handle different client devices such as accessing content via a Web browser or through a cell phone.

There are two types of generation templates. Detail view generation templates provide a view of a single piece of content. For example, the detail view of an article might show the title, author, and body. Summary view generation templates typically show a list of one-line descriptions about each piece of content included in the summary with a hyperlink to the detail view of each piece of content. Figure 1-2 on page 7 illustrates how summary and detail templates are used to generate Web pages.
In this example, the summary view for a set of articles might show the headlines with links to the detail article view. Summary views can be generated for all elements within a content type, or all elements within a folder. The folder can be a fixed folder within a content type, or a folder defined by a search.

Note that generation templates are generally thought of as generating static pages. However, that is not necessarily the case. You can use Web Content Publisher to create static pages or JSPs as output. In this way, you can include dynamic information on pages generated with templates.

**Publishing**
Publishing environments do not publish directly to the production server. Staging servers must be used to view and manage content before it is available to the public. Therefore, Web content management systems must support publishing content to a remote server. This requires that the Web content management system has some method of transferring files from one machine to another, such
as FTP. The transferred files must also map from a directory structure on the transmitting server to a receiving server.

Typically a publishing environment contains at least a development server and a production server. Publishing directly into production is not recommended.

Publishing may need to be scheduled. Content may need to appear on a Web site as a logical group, such as an edition of a newspaper. In the scenario, Tara WebMaster created a full edition of the Web site and scheduled publishing to begin at 3 a.m.

**Versioning and editioning**
A multiple user publishing environment requires file-level locking to avoid users modifying content simultaneously.

The version control in Web content management systems is similar to managing source code during software development. Locking is required to avoid multiple developers modifying the same piece of code. A team leader consolidates all source code together, testing is performed, and the package is migrated into production.

Web content management systems also require the ability to create editions. An edition is a snapshot of all the Web content. An edition is created when an editor receives many contributions from authors and needs to create a consolidated view of the Web site. In the scenario, Tara WebMaster consolidates all contributions and creates an edition to publish to the production server.

**Note:** There is no current support for external version control. CVS support is limited to import and export through Web Content Publisher from WebSphere Studio Application Developer.

**Workflow**
Content must be requested, reviewed, accepted, and approved before it can be published to the Web. The business processes that define how content is published is the publishing workflow.

In the above scenario, Joe SportsEditor was able to reject Greg ContentContributor’s article. This was because the workflow was implemented for their organization to allow Joe to veto a story.

**Administration**
Each Web content management system must manage users, user permissions, groups, and security. In the above scenario, Joe SportsEditor did not have proper permission to submit content directly to the production server. The scenario
would likely not allow Greg ContentContributor to create or modify presentation templates because he is not adept at HTML.

1.3 Installation

This section describes how to install WebSphere Portal with Domino and the Web Content Publisher. This makes it possible to leverage the portal's ability to provide real-time messaging via Sametime, Collaborative Places, and Web content management via Web Content Publisher.

This installation describes a scenario where WebSphere Portal is installed with Domino providing the authentication through its LDAP server. Additional steps are also used to install Web Content Publisher that may be omitted, if not required.

**Important:** If Web Content Publisher is not installed initially with WebSphere Portal, difficulties may occur if you attempt to integrate it later. If there is any possibility that your organization will use Web Content Publisher, please perform the additional steps. This will not detract from the performance of your Domino server, and will provide a risk-free benefit.

1.3.1 Patched rt.jar file

As of this writing a patched rt.jar file is required for the installation of the WebSphere Portal in 1.3.11, “Install WebSphere Portal” on page 76. You will need to obtain this from IBM support.

1.3.2 Remove Lotus Notes clients

If you are installing this on a machine where you are currently using your Notes client, you can use the following procedure to remove Notes before installing WebSphere Content Publisher and install another copy afterwards. Note that this will result in having two copies of Notes.

If you have any questions about this process, please contact your Notes system administrator.

1. Make a backup of your Lotus Notes Data directory (typically \lotus\notes\data or C:\Notes\data).
2. Make sure your ID file is in that backup. The ID file is used to uniquely identify the user and usually has an .id suffix. If not copy into the backup data directory.
3. Record your IBM Notes Server name.
4. Uninstall Lotus Notes and remove the directory it was installed in. This is typically C:\Notes or C:\Lotus\Notes.
5. Do the WebSphere Portal install described in this document.

Once the WebSphere Portal install has completed, you may reinstall the Lotus Notes client. To avoid overwriting the Domino install used for Portal Server, you must:

- Specify a separate location from the Notes that was installed for Portal Server. Do not use C:\Notes or C:\Lotus\Notes.
- Specify a different folder for the Program menu. Do not use Lotus Notes.

Once you have completed the reinstall, you may restore Notes.

1. Copy the contents of the backup Data directory made in Step 1 on page 9 to the Data directory for your new install.
2. Start Notes and configure it to your Mail Server.

Attention: Make sure you do not try to use two Lotus clients pointing at different servers at the same time. For example, do not have a Domino Administrator open against the WebSphere Content Publisher Domino Server and then try to start Notes against the IBM Mail server.

1.3.3 Install DB2, IBM HTTP Server and WebSphere Application Server

The first step of our installation is to install the following components:

- DB2
- IBM HTTP Server
- WebSphere Application Server

WebSphere Application Server is installed before installing Domino Application Server, because keys used to create single sign-on communication between them must be created by WebSphere Application Server prior to the install of Domino.

The installation is identical to 5.2, “Installing WebSphere Portal with SecureWay using the Setup Manager” in IBM WebSphere Portal V4.1 Handbook Volume 1, SG24-6883, except step 6 in 5.2.4, “Secureway LDAP” in that volume when components are being selected. Only DB2, IBM HTTP Server, and IBM WebSphere Application Server should be selected. Do not select Web Content Publisher or Domino Application Server at this time.
The selected components should appear as shown in Figure 1-3.

![Selecting Components](image)

**Figure 1-3** Select components DB2, WebSphere and IBM HTTP Server

The installation values will be identical for the various components. The final display summary in step 7 in 5.2.10, "WebSphere Portal" of *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883 should appear as shown in Figure 1-4 on page 12.
Once the installation process has completed, test that WebSphere Application Server is working correctly using the snoop servlet described in step a in 5.2.11, "Installation Procedure" in IBM WebSphere Portal V4.1 Handbook Volume 1, SG24-6883.

**Tip:** Make sure that your browser cache has been cleared before any testing throughout this installation process.

### 1.3.4 Generating keys in WebSphere Application Server

WebSphere Application Server will provide single sign-on between itself and Domino Application Server by sharing Lightweight Third Party Authentication (LTPA) tokens. LTPA tokens contain user data, expiration time, and a digital
signature that is signed with a private key of the authenticating user. They are stored as encrypted cookies.

A key for decrypting the cookie is shared by WebSphere Application and added to Domino Application Server.

This following describes how WebSphere Application Server creates the key that will be shared by Domino:

1. Click Start -> Settings -> Control Panel. Double-click Administrator Tools. Double-click Services. Check to see that IBM WS AdminServer 4.0 has started. If it has not, right-click IBM WS AdminServer and select Start.

2. Start the WebSphere Application Server by clicking Start -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console.

3. Select Console -> Security Center. You will see a window similar to Figure 1-5 on page 14.

5. Click Generate Keys... button. You will see a prompt asking for an LTPA password similar to Figure 1-6 on page 15.
6. Enter the password. Click **OK** and the LTPA password window will close.

7. Click **Export Key**... You will see a window similar to Figure 1-7.

8. Select a location and file name. For our example, we selected the C:\ directory and the file name DOMWAS.key. Click **Save**.

9. Reboot the machine.

The key file DOMWAS.key is required during the installation of Domino Application Server. Now that it is generated, we can continue to install Domino Application Server and other components.

**1.3.5 Install Domino components and Web Content Publisher**

We will now install the following Domino components:

- Domino Application Server
- Lotus Architect
- Lotus Workflow
This section includes the additional steps required to install Web Content Publisher. While this is optional, it is recommended that Web Content Publisher be installed at this time. If your organization has any interest in using it, follow the additional installation steps.

1. Start the installation process by inserting CD1 and executing the install.bat file.

2. Read and select I accept the program license agreement. Click Next.

3. Enter your license key. Click Next.

4. Select Standard Installation for the install type and click Next.

5. Leave the response file location empty and click Next.

6. Select Web Content Publisher. This will automatically select IBM HTTP Server (previously installed), WebSphere Application Server (previously installed), and WebSphere Personalization.

7. Select Domino Application Server. A Domino Application Server is needed by WebSphere Content Publisher to run applications such as Lotus Workflow and LDAP. Select Lotus Workflow and Lotus Architect. Lotus Workflow will install itself on the local machine and Lotus Architect will install its client on the local machine. Do not install WebSphere Portal Server at this time. After these selections, your window should look similar to Figure 1-8 on page 17 and Figure 1-9 on page 18 (after scrolling). Click Next.

**Important:** The WebSphere Content Publisher Publish Servers cannot be installed at the same time as the WebSphere Content Publisher Server. If you need to install Publish Servers, please run the install again after installing the WebSphere Content Publisher Server and select the Publish Servers that you want installed.

You should not install the WebSphere Content Publisher Personalization Publish Server if you are installing WebSphere Portal Server. The Portal Content Organizer component of WebSphere Portal Server will install the WebSphere Content Publisher Personalization Publish Server. Do not re-install WebSphere Content Publisher Server and Samples over the top of an existing install without backing up the WCM database. The re-install will reset the databases.
Figure 1-8   Selecting Domino components to install
8. A window will display a list of all previous installed components. Click **Next**.

9. The system will now check previous installations. Note that IBM HTTP Server, Global Security Toolkit, WebSphere Application Server, and WebSphere Application Server Fixpack 2 are already installed and will take no action. Click **Next**.

10. Click **No** for WebSphere Application Server Security enabled. Click **Next**.

11. Enter the administrator ID, `wasadmin`, with `wasadmin` as the password for the administrator ID. Click **Next**.
12. Leave the default WebSphere Portal for the application server for Personalization server to run on. Click Next. You will see a window similar to Figure 1-10.

![Figure 1-10 Select Domino configuration type](image)

**Figure 1-10** Select Domino configuration type

13. Accept the default Web Content Publisher for the Domino Server configuration type. Click Next. You will now see a window similar to Figure 1-11 on page 20.
14. Accept the default Domino Application Server for the default Domino Server type of install. Click **Next**. You will now see a window similar to Figure 1-12 on page 21.
15. Accept the defaults. This defines the installation path for the Domino Server. Click **Next**. You will now see a window similar to Figure 1-13 on page 22.
16. Enter passwords for certifier password and Domino administrator password, and confirm them. These are passwords used to administer and manage the Domino server. Ensure that the domain name, certifier organization, server name and host name are correct. The server name should be the name of the node you are installing on. The host name should be the fully qualified domain name for the installation machine. Accept the remainder of the defaults. In our example, we used the password password. Click Next. You will see a window similar to Figure 1-14 on page 23.

Tip: The Domino Administrator account will be created with a user ID and Shortname of dadmin. When you see this user ID further in the installation, it is referring to the Domino Administrator account.
17. Leave the defaults. Select **Web Server**, **DIIOP** and **LDAP**. Ensure that **Configure SSO Support at this time** is set to Yes. Selecting Web Server will utilize the HTTP server from Domino. Domino Directory Services also provides an implementation of LDAP. This must be selected if you intend doing authentication and authorizing through Domino. Click **Next**. You will see a window similar to Figure 1-15 on page 24.
Figure 1-15  HTTP Server ports for Domino

18. Accept the default port. Port 80 will not be used by Domino because IBM HTTP Server is currently using it. Note that you may not see this window if you did not install the Web Server in step 17 on page 23. Click Next. You will see a window similar to Figure 1-16 on page 25.
19. Enter `C:\DOMMAS.key` in the LTPA File field. This is where the key file that was created using the WebSphere Administration Console is used (see Figure 1-7 on page 15). Enter the LTPA password and the token domain. In our example, we used our domain `itso.ral.ibm.com`. This domain must match the domain specified in step 3 on page 13. Click **Next**. You will see a window similar to Figure 1-17 on page 26.
Figure 1-17  Domino Client install location

20. Accept the default locations for the Domino clients to be installed. Click Next. You will see a window similar to Figure 1-18 on page 27.

Note: The default token domain may appear as above, preceded by a period. This will be accepted by the installation process.

The following steps will be performed. If you are not installing Web Content Publisher, you will not see these windows.
21. Select **DB2** as the database for Web Content Publisher. Web Content Publisher will use DB2 to store user content. Click **Next**. You will now see a window similar to Figure 1-19 on page 28.
22. Enter the db2 administrator's user ID and password. In our example, we used the user ID of db2admin with the password db2admin. This allows WebSphere Content Publisher to create new databases in DB2. Click Next. You will now see a window similar to Figure 1-20 on page 29.
23. Select **Local** for the connection type to Lotus Workflow server. Click **Next**.

24. A disk space check will be displayed. Click **Next** and the install will begin. During the install you may see Domino pop up. Do not close or kill any of these windows as they are required by Setup Manager to do the install.

**Note:** The WebSphere Content Publisher install might report a problem, but it is likely OK. If the install hangs at 95-99% complete, then check the Services window and if the Admin Service is stopped, restart it, and the install will complete. After WebSphere Content Publisher was installed (silently), the Setup Manager tried to stop and start the WS Admin Server and it failed.

**Note:** If the WebSphere Content Publisher install hangs at 50% complete, kill the Setup Manager by using Ctrl+C in the command window where install.bat was run. Uninstall WebSphere Content Publisher and Lotus Workflow Architect using Add/Remove programs. Reboot the machine and restart the install with WebSphere Content Publisher.
Once the install of WebSphere Content Publisher has completed, you will be guided through the installation of Lotus Workflow 3.0 Architect. You will now see a welcome window to install Lotus Workflow 3.0a Architect (Figure 1-21).

![Lotus Workflow welcome window](image)

**Figure 1-21  Lotus Workflow welcome window**

25. Click **Next**. You will see a window similar to Figure 1-22.

![Destination Directory](image)

**Figure 1-22  Destination Directory**
26. Accept the default Notes Program directory and click **Next**. You will see a window similar to Figure 1-23.

![Select destination to install Architect](image1.png)

**Figure 1-23** Select destination to install Architect

27. Click **Next**. You will see a window similar to Figure 1-24.

![Lotus Workflow Architect program folder](image2.png)

**Figure 1-24** Lotus Workflow Architect program folder
28. Accept the default program folder and click **Next**.

![Setup Complete](image)

*Figure 1-25  Workflow installation is complete*

29. Allow Lotus Workflow 3.0 Architect to install. Once it has completed, click **Finish**. You will see a window similar to Figure 1-26. The installation is complete. Click **OK**.

![Installation Complete](image)

*Figure 1-26  Installation is complete*

### 1.3.6 Configure Domino Administration client

This section describes how to configure the Domino Administrator client that allows us to manage and configure the Domino server. This applies for both Domino LDAP and WebSphere Content Publisher installations. This step must be performed by anyone who will administer the Domino Application Server.
1. Click **Start -> Lotus Applications -> Lotus Domino Server**. This will start the Domino Server without using the services window. Do not start using the services window.

2. Click **Start -> Programs -> Lotus Applications -> Lotus Domino Administrator**. This will start the Domino Administrator. You will see a window similar to Figure 1-27.

![Figure 1-27 Welcome window for configuring Lotus Notes client](image)

3. The Lotus Notes Client Configuration window is displayed. Click **Next**. You will see a window similar to Figure 1-28 on page 34.
4. Select **I want to connect to a Domino server** and click **Next**. You will see a window similar to Figure 1-29.

5. Select **Set up a connection to a local area network (LAN)** and click **Next**. You will see a window similar to Figure 1-30 on page 35.
6. Enter your server name in the Domino server name field. In our example, we entered m23wpn62/itso.ral.ibm.com. Click **Next**. You will see a window similar to Figure 1-31.

7. Select **Use My Name as identification**. Type your Domino Administrator name. This was Domino Admin, and was specified in step 15 on page 21. Click **Next**. You will see a window similar to Figure 1-32 on page 36.
8. Click **Next**. You will see a window similar to Figure 1-33.

9. Select **I don't want to create an Internet mail account**. Click **Next**. You will see a window similar to Figure 1-34 on page 37.
10. Select **I don't want to connect to a news server**. Click **Next**. You will see a window similar to Figure 1-35.

11. Select **I don't want to connect to another directory server**. Click **Next**. You will see a window similar to Figure 1-36 on page 38 that determines whether you will connect through a proxy server.
12. Select the choice that is appropriate for your installation. If you are unsure, ask your system administrator. For our example, we selected **I do not connect to the Internet through a proxy server**. Click **Next**. You will see a window similar to Figure 1-38 on page 39.

If you select that you are connecting to the Internet through a proxy server, then you will have an additional window shown in Figure 1-37 on page 39. Fill it out appropriately and click **Next**.

**Tip:** If your installation requires a proxy server, you may obtain the necessary information through the Microsoft Internet Explorer browser by choosing **Tools -> Internet Options...** Open the **Connections** tab and click **LAN Settings**.... This will also indicate whether or not you are using a proxy server.
13. Select **Connect over local area network or cable modem** and click **Next**. You will see a window similar to Figure 1-39 on page 40.
14. You should receive a notice that you have successfully set up Lotus Notes. Click Finish. You will see a window similar to Figure 1-39.

![Figure 1-39 Successful install of Lotus Notes]

15. You will be prompted for the Domino Admin password. Enter the password and click OK.

16. The server will create your address book and you will see a note stating that Notes setup is complete. Click OK.

You may receive the message, Notes Error - Specified Command is not available from the Workspace. You can ignore this error message. Click OK.

17. Close the Domino Administrator.

1.3.7 Configure a workflow for Web Content Publisher

The following describes how to configure a workflow for Web Content Publisher. If you are not installing Web Content Publisher, you can skip this section.
Configure Workflow Architect
This section describes the configuration of Lotus Workflow Architect. Perform the following instructions:

1. Click Start -> Programs -> Lotus Workflow 3.0a Architect -> Lotus Workflow 3.0a Architect. This will start the Lotus Workflow Architect program.

2. Select File -> Open Databases. A window will appear as shown in Figure 1-41.

![Figure 1-41 Importing data sources](image)

3. Click New at the upper left of the Data Sources window. You will see a window similar to Figure 1-42.

![Figure 1-42 Creating the WebSphere Content Publisher profile name](image)

4. Enter WCP as the Profile name. Click OK.
5. Select **Design Repository**. It is located under Data Source Type (Figure 1-41 on page 41). Click **Browse**. You will see a file-based repository as shown in Figure 1-43.

![Select a Design Repository database](image)

*Figure 1-43 Selecting a design repository database*

6. Under the Server drop-down menu, select your server. If prompted, enter your password, which is the Domino Administrator's password. For our example, we used `password`. If your server name is not listed in the drop-down, you must type it in manually (for example, m23wpn62/itso.ral.ibm.com).
7. Under the Database menu (shown in Figure 1-44), select **LWF Design Repository R3.0** and click **OK**. Use the Up arrow if you do not see this item listed.

8. Repeat the above process for the Data Source types: Application database, Process Definition database, and Notes Organization Directory, which will match up with LWF Application R3.0, LWF Process Definition R3.0, and LWF Organization R3.0 respectively (see Figure 1-41 on page 41). Your window should look similar to Figure 1-45 on page 44 with check marks beside Design Repository, Application database, Process Definition database, and Notes Organization Directory, respectively.
9. Click **OK**. We will now import the workflow files.

10. Select **File -> Import**. This will open a file window as shown in Figure 1-46.

11. Click **Browse** to locate the LWF file that is in \wcp\wcp\wfp\process\SimpleChangeProcess.lwf on CD 9 and click **Open**. Click **OK**. You should see a flowchart similar to Figure 1-47 on page 45.
12. From the menu bar, select **File -> Save Process**.... If you get a warning message saying Process SimpleChange Process has not been modified. Do you want to save it anyway?, click **Yes**.

13. Select **File -> Activate Process**.... You will see a window similar to Figure 1-48 on page 46.
14. Accept the defaults and click **OK**.

15. Repeat steps 10 through 14 to import the other two workflow processes provided by Web Content Publisher:
   - SimplerChangeProcess.lwf
   - SimplestChangeProcess.lwf


### Configuring the workflow process

To configure the workflow process, perform the following steps:

1. Click **Start -> Programs -> Lotus Applications -> Lotus Domino Administrator**. If prompted for the Domino Admin password, enter the password and click **OK**.

2. From the top-left menu, click **File -> Tools -> Switch Id**. Navigate to the `lotus\domino\data` directory and select **WCPAdmin.ID** as shown in Figure 1-49 on page 47.
Chapter 1. Web content management

3. Click **Open**. A password window will appear. Enter **password** as the password and click **OK**.

4. Click **File -> Database -> Open**. You will see a window similar to Figure 1-50.

5. Select your server from the **Server** menu as shown in Figure 1-50. Scroll to locate and select the **LWF Application R3.0** database and click the **Open** button. You may see some notifications to trust signers or certificates or to create cross-certificates. Click **Yes** or **Trust Signer** for all notifications. An example is shown at Figure 1-51 on page 48.
6. Click the **Administration** view in the top-left portion of the window. Select **File -> Open Server** from the top-left menu pull-down. You will see a window similar to Figure 1-52.

![Figure 1-52 Selecting our server to administrate](image)

7. You should not be connected to the Local server. Select the host name you created (not Local) as shown in Figure 1-52. For our example, we entered `m23wpn62/itso.ral.ibm.com`. Click **OK**. You will see a window similar to Figure 1-53 on page 49.
8. Click the **Files** tab (located beside People and Groups). A list of databases are listed to the right under Title and Filename such as Administration Requests, Java AgentRunner, etc.

**Tip:** If you do not see a list of files, close and reopen the Lotus Domino Administrator.

9. In the list of database files, double-click **LWF Application R3.0** (application_1.nsf). If the system asks you whether you trust the signer and accept the certificates, respond with Yes if a cross-certificate is requested. If necessary, press Esc. You should see a window similar to Figure 1-54 on page 50.
10. Return to the Administration tab and click **LWF Organization R3.01-1** Workgroups view. This will ensure you are working with the Organization - Workgroups database. See Figure 1-55 on page 51.
11. On the left pane, select Administration -> Cache. On the top pane, click Update Cache. If you are prompted, trust the signer. If a message appears, click OK.

12. Click the LWF Application R3.0 database view. You should see the three processes in a window similar to Figure 1-56 on page 52.
13. Close the LWF Application R3.0 database by exiting Lotus Domino Administrator. Messages may display about a window that is not closed and a message about removing anyway. Click No and continue.

### 1.3.8 Configuring WebSphere Application Server security

We will now configure WebSphere Application Server's security. By enabling security, WebSphere will begin to use Domino LDAP for authentication.

1. Click Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Start Admin Server to ensure the Admin Server is running. This will open a command prompt. Wait until it has disappeared before continuing. If it disappears immediately, the Admin Server may already be running.
2. Click **Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console.** You should see a window similar to Figure 1-57 on page 53.

![WebSphere Advanced Administrative Console](image)

**Figure 1-57  WebSphere Advanced Administrative Console**

3. Select **Console -> Security Center.** You will see a window similar to Figure 1-58 on page 54.
4. Select the **General** tab, and then check **Enable Security** as shown in Figure 1-58.

5. Select the **Authentication** tab. You will see a window similar to Figure 1-59 on page 55.
6. Modify the items in the lower portion of the window. Select the **LDAP** button. In the Security Server ID field, enter *dadmin*, which is the short user ID for the Domino Administrator. Enter the Domino Administrator's password in the password field. Enter your fully qualified host name in the host field. Select **Domino 5.0** as the directory type. Leave all other fields set to default and click **OK**. If you are prompted, enter the LTPA password, which we had configured as password. The message *The changes will not take effect until the admin server is restarted* will appear. Your window should look similar to Figure 1-59. Click **OK**.

7. Close the WebSphere Advanced Administrative Console.

8. Click **Start -> Settings -> Control Panel**. Double-click **Administrative Tools**. Double-click **Services**. Right-click **IBM WS AdminServer** and select **Stop**. Once the process has stopped, right-click **IBM WS AdminServer** and select **Start**.
9. Click **Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console**. A request for a password is now required. Enter **dadmin** as the user identity and the Domino Administrator's password (the default during the install was **password**) as the user password. Click **OK**. The Administrative Console should now appear. This verifies that WebSphere Application Server is using Domino as its LDAP source.

If the server was requested to start but a message displays saying the service did not respond in a timely fashion, this usually means Domino has problems or is not running or it is taking longer than the normal waiting period. Wait a while and refresh the Services window to see if it is started.

### 1.3.9 Verify the Web Content Publisher install

Web Content Publisher should now be available as a Web module. We will now verify that the install has worked correctly.

Web Content Publisher does not require WebSphere Portal to run and will be installed later. However, you will notice that a WebSphere Portal is listed when viewing the application servers in WebSphere Advanced Administrative Console. This is because Personalization (which is required to be installed by Web Content Publisher) creates this application server. The full WebSphere Portal install is not completed until later.

1. Ensure the following services are running by clicking **Start -> Settings -> Control Panel**. Double-click **Administrative Tools** and double-click **Services**.
   - Lotus Domino Server (LotusDominodata)
   - IBM WS AdminServer 4.0

   **Tip:** When starting WebSphere Content Publisher, Lotus Domino Server (LotusDominodata) must be running before IBM WS AdminServer 4.0 is started. This is because IBM WS AdminServer relies on Lotus Domino Server to provide the LDAP service to enable WS AdminServer security.

   **Tip:** It is important to note that Domino Server may appear to be started in the Services window, but has not yet been completely initialized and therefore not available. When the Lotus Domino Server is started, a command prompt will appear with information on the server's status. Ensure that it looks like Figure 1-60 on page 57 where it says that HTTP Server is running and LDAP Server has started.
2. If the WebSphere Administrator's Console is not open, click **Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console**. It will ask for a password. The User identity is dadmin and the password is the Domino Administrator's password.

3. Expand **WebSphere Administrative Domain -> Nodes -> <your node name> -> Application Server**. Right-click **WebSphere Portal** and select **Start** if it is not running (note that WebSphere Portal is running in Figure 1-61 on page 58).
4. From the IE browser, enter the URL

   http://<your fully qualified host name>/wps/wcp/index.jsp

5. You should see a window similar to Figure 1-62 on page 59.
6. Enter the user ID rob and password rob and click the Login button. The user rob was added during the configuration of Lotus Workflow. You should now see a window similar to Figure 1-63 on page 60.
Troubleshooting

If you did not get Web Content Publisher to install correctly, consider one of the possible problems:

- Reboot the system before doing any debugging.
- Make sure that Domino Server was running before WS Admin Server service.
- Ensure that the WebSphere Portal Application Server is started. This was performed in step 3 on page 53.
- Verify SSO configuration:
  
  a. Try snoop by opening http://<fully qualified host name>/servlet/snoop. Type a user ID of dadmin and a password of password. Make sure the Default Server Application Server is started.
  
  b. In the same browser session, type http://<fully qualified host name>:8080/Process_Definition_1.nsf. You should not be prompted for another sign-on. If you are, then SSO is not set correctly.
     - Look in the WAS_HOME\bin stdout.txt, stderr.txt directory.
Check the Troubleshooting section of WebSphere Content Publisher Readme in the wcp directory of CD9.

Log files for installs using WPO Setup Manager are most likely found in the c:\program files\IBM\WPO directory with a filename such as setup*.log. Old logs are in the logs directory. The log file lists the commands being executed. You can also access the file during install by clicking the Setup Log button on the Display Summary. Output of individual commands are specified in the setup*.log, usually the c:\winnt\temp\runcommand directory.

1.3.10 Configure Domino for WebSphere Portal

Before installing WebSphere Portal, it is necessary to make manual configuration changes to Domino. The following describes what changes are required:

1. Click Start -> Programs -> Lotus Applications -> Lotus Domino Administrator to start the Domino Administrator. You will be prompted for a password. Enter the password for the appropriate ID and click OK.

2. If you are not using the Domino Administrator ID, switch to it. Click File -> Tools -> Switch ID… This will open a window similar to Figure 1-64. Navigate to the C:\Lotus\Domino\data folder and select user.id. This is the Domino Administrator's ID. Click Open and enter the password.

![Choose User ID to Switch To](image)

Figure 1-64 Switch user ID to Domino Admin using the user.ID file

3. Click File -> Open Server. You will see a window similar to Figure 1-65 on page 62.
4. Select your server from the drop-down menu. Do not select the local server. Click **OK**.

5. Go to the Administration view. Click the **Configuration** tab. You will see a window similar to Figure 1-66.

6. From the navigation on the left, expand Server and then click **Current Server Document**.

7. Click **Internet Protocols** tab. Enter the fully-qualified host name in the Host name(s) field. In our example, we entered `m23wpn62.itso.ral.ibm.com` as
shown in Figure 1-66 on page 62. Click **Save and Close**. This will save the document, but the document will not close.

![Domino server configuration](image)

**Figure 1-67** Domino server configuration

8. Click **Configurations** in the left-hand pane (Figure 1-67) underneath the Server list.

9. Click **Add configuration** in the right-hand pane. You will see a window similar to Figure 1-68 on page 64.
10. Select **Yes** to use these settings as the default settings for all servers.

11. Click the **LDAP** tab. You will see a window similar to Figure 1-69 on page 65.
12. Click **Choose Fields that anonymous users can query via LDAP** button. This will display a pop-up window shown in Figure 1-70.
13. Click **Show Fields**. From the Fields in form: Person pane, select **MailFile** and **MailServer**. Click **Add** to add them to the already selected list. See Figure 1-70 on page 65.

14. Click **New**. A pop-up window titled New Field will appear (Figure 1-71).

```
Figure 1-71   Adding a new field to LDAP
```

15. Enter **HTTP_HostName** and click **OK**.

16. Click **OK** on the LDAP Field list window.

```
Figure 1-72   Allowing LDAP users write access
```

17. In the Allow LDAP user write access field at the bottom of the window, choose **Yes**. Click **Save and close**.
18. Open the **People & Groups** tab. Click **Groups** in the left-hand pane.

19. Click **Add Group** in the right-hand pane. You will see a window similar to Figure 1-74 on page 68.
Figure 1-74 Add the wpsadmins group to Domino

20. Enter wpsadmins in the Group name field. Click **Save and Close**.
Figure 1-75  Selecting the Register button

21. Open the **People & Groups** tab. On the right-hand side of the tool bar, open the **Tools** menu, open the **People** menu and click **Register**.... You will see a window similar to Figure 1-76.

Figure 1-76  Selecting the certifier ID
22. Select the cert.id file in C:\Lotus\Domino\data and click **Open**.

23. A password prompt will appear. Enter the certifier’s ID as specified during the install of Domino. We used *password*. Click **OK**. A warning may pop up claiming that the current certifier ID contains no recovery information. Click **Yes** and continue.

24. Select the **Advanced** check box in the top-left corner. Leave the first name blank and enter **wpsadmin** as the last name. Also ensure that the short name is **wpsadmin**. Enter **wpsadmin** as the password. Select **Set internet password** option. Enter an Internet address and Internet domain based on your host name. See Figure 1-77. The password must be wpsadmin for the install to work properly.

25. Click **Groups**. You will see a window similar to Figure 1-78 on page 71.
26. Select wpsadmins and click Add. Click Add Person.

27. Click the Basics button on the left of the Register Person window. Repeat the process using wpsbind instead of wpsadmin. Ensure the password is wpsbind and that Set Internet password is selected. Ensure that the short name is also wpsbind. The password must be wpsbind for the install to work properly. Add wpsbind to the wpsadmins group as described in step 25 on page 70. Click Add Person when you are done.

28. Click Register All. This will now create the wpsadmin and wpsbind users and make them available to the Domino LDAP system. WebSphere Portal requires these users to install the portal.

29. You will see a pop-up window stating All 2 people registered successfully! Click OK to continue. Close the Add Person window.
30. In the Administration view, click the **Files** tab. There is a names.nsf file located under the Filename column. Right-click it and select **Access Control** -> **Manage** as shown in Figure 1-79. Next, you will see a window similar to Figure 1-80 on page 73.
31. Click **Add**. You will see a window similar to Figure 1-81.

32. Click the blue person button to see a window titled Names (Figure 1-82 on page 74).
33. Select the host name address book from the top-left pull-down menu. Select the `wpsadmin` user from the left-hand pane and click Add. Click OK.

34. Select the `wpsadmin/itso.ral.ibm.com` user in the Access Control List window. In the User type pull-down menu, select Person. In the Access
pull-down menu, select Manager. Leave the Delete documents selected. Ensure each role in the Roles menu is checked (Figure 1-83 on page 74).

35. Click Add… button. This will pop up an Add User button. Click the blue person button and select wpsadmins group as done previously in step 32 on page 73. Click Add and click OK.

36. In the Access field, select Manager. Ensure all roles are selected and Delete documents is selected as shown in Figure 1-84.

![Figure 1-84 Adding permissions for wpsadmins group]

37. Click OK.

38. In the Command Prompt where Domino server was started, type quit and press Enter. Restart the Domino server from the menu. This will allow all changes to take place.

**Verify users have been added to Domino LDAP**

We will now verify that the wpsadmins group, wpsadmin user, and wpsbind user required by WebSphere Portal have been successfully added to Domino’s LDAP.

1. Click Start -> Programs -> Accessories -> Command Prompt.

2. Navigate to the c:\lotus\Domino directory. Enter the command:

   ```
   Ldapsearch -h hostname/domainName cn=wps*
   ```

   where hostname/domainName is your fully qualified Domino Server name.

---

Chapter 1. Web content management 75
3. You should see entries similar to Figure 1-85. The certificate field will not be the same, but ensure that the wpsadmin and wpsbind users and wpsadmins group are created.

```
C:\Lotus\Domino>Idapsearch -h m23ump62/itso.rale.ilm.com cn=ups+CN=wpsadmin
on=wpsadmin
mail=wpsadmin@itso.rale.ilm.com
objectclass=top
objectclass=groupOfNames
objectclass=dominoGroup
member=CN=wpsbind,0=itso.rale.ilm.com
member=CN=wpsadmin,0=itso.rale.ilm.com

CN=wpsbind,0=itso.rale.ilm.com
 on=wpsbind
 mail=wpsbind@itso.rale.ilm.com
 objectclass=top
 objectclass=person
 objectclass=organizationalPerson
 objectclass=inetOrgPerson
 objectclass=dominoPerson
certificate=8D0D01F02 B80D80D1 120B161E GM02E7B4
 on=wpsbind
 mail=domain=itso.rale.ilm.com

mail server=CN=m23ump62,0=itso.rale.ilm.com
 mail file=mail\wpsbind


Figure 1-85   LDAP search
```

Domino has now been configured for WebSphere Portal installation.

### 1.3.11 Install WebSphere Portal

The final process in our installation is to install WebSphere Portal.

**Replace rt.jar in WebSphere Application Server**

Prior to installing WebSphere Portal, we must perform the following:

1. Contact IBM support and obtain the latest copy of rt.jar for WebSphere. If you do not do this you may encounter an error that looks like this:
Chapter 1. Web content management

(Sep 23, 2002 5:00:33 PM), install, com.ibm.wps.install.LdapCheckPanel,
msg2, Calling LDAP check with itso-0n5i4hw5xh.dominotest.com:389;
cn=wpsadmin{o=dominotest;cn=wpsbind,o=dominotest;cn=wpsadmin,o=dominotest;c
n=wpsadmins}
Checking for 'o=dominotest'
Checking for 'cn=wpsbind,o=dominotest' javax.naming.CommunicationException:
Socket closed [Root exception is java.net.SocketException: Socket closed];
remaining name 'cn=wpsbind,o=dominotest'
(Sep 23, 2002 5:00:33 PM), install, com.ibm.wps.install.LdapCheckPanel,
err, Code 2
This file will be used temporarily for the installation, then replaced with the
original.

2. If the WebSphere Administrative Console is open, close it.

3. Click Start -> Settings -> Control Panel. Double-click Administrative
   Tools. Double-click Services. In the Services window, right-click WS Admin
   Server 4.0 and select Stop (if it is not already stopped).

4. Rename c:\WebSphere\AppServer\java\jre\lib\rt.jar to rt.old.

   Tip: If you cannot rename rt.jar, close any other programs that might be
   related to WebSphere, then try rebooting your server.

5. Copy the patched rt.jar file to c:\WebSphere\AppServer\java\jre\lib\rt.jar.

6. Return to the Services window. Right-click Lotus Domino Server
   (dominodata) and select Start. This will execute a Command Prompt. Ensure
   that it has run to completion as shown in Figure 1-60 on page 57.

7. Right-click WS Admin Server 4.0 and select Start.

Disable security in WebSphere Application Server
WebSphere Application Server security will be disabled.

1. Click Start -> Programs -> IBM WebSphere -> Application Server V4.0 ->
   Administrator's Console. Password prompt will request a user identity and
   user password. Use dadmin and password if using the Domino
   Administrator's default password.

2. Select Console -> Security Center... This will display a window similar to
   the one shown in Figure 1-57 on page 53.

3. Deselect Enable Security as shown in Figure 1-86 on page 78. Click Apply.
   A warning message will pop up saying that changes will not take effect until
   the admin server is restarted. Click OK.
Figure 1-86  Disabling security in WebSphere Application Server

4. Click **OK** in the Security Center and exit the WebSphere Administrator’s Console.

5. Return to the Services window. Stop and restart the WS Admin Server.

**Install Portal**

Perform the following steps to install WebSphere Portal:

1. Insert Disk 1 into the CD-ROM drive. The installer should begin to run.

2. Accept the license, enter the license key, and select a **Standard** install. These steps are identical to those in 5.2.4, “Secureway LDAP” in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883. Continue to step 7 in that volume, where components are being selected if necessary.

3. In our install, select only **WebSphere Portal**. This will automatically include WebSphere Personalization, WebSphere Application Server, and IBM HTTP Server. WebSphere Application Server and IBM HTTP Server were already
installed previously and will not be installed again. Ensure that Lotus Collaborative Places and Components is not selected (it will be by default). You should have WebSphere Portal, WebSphere Portal, Productivity Portlets, and Portal Server checked. You should have checked WebSphere Personalization, WebSphere Personalization, Personalization Server, WebSphere Application Server (Fixpack2 and WebSphere Application Server) and IBM HTTP Server. Your window will look similar to Figure 1-87. Click Next.

4. You will see that some products have already been installed, similar to Figure 1-88 on page 80. In this particular scenario, Global Security Toolkit,
IBM HTTP Server, WebSphere Application Server, Personalization Server and others had already been installed in previous steps. Click **Next**.

![Checking previous installations](image)

Figure 1-88 Checking previous installations

5. Select **No** for WebSphere Application Server Security enabled. Security was shut off in “Disable security in WebSphere Application Server” on page 77. Security is disabled for the WebSphere Portal install.

6. Choose **Typical** for the installation type and click **Next**.

7. Choose **Database and LDAP Directory** and click **Next**.

8. Choose **Later** for enabling security configuration as shown in Figure 1-89 on page 81. We will configure security after our install; you should not do it now. Click **Next**.
9. Allow the default values for the Server configuration as shown in Figure 5-15 in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883, modify the proxy host or port if necessary, and click **Next**.

10. Select **Lotus Domino Application Server** as the LDAP server. Update User_DN to cn=wpsadmin,o=<yourDomainName>. You must use the values from the ldapsearch performed in “Verify users have been added to Domino LDAP” on page 75. The password to be entered is wpsadmin. Leave Suffix blank and ensure LDAP port number is 389. Your window should look similar to Figure 1-90 on page 82.
11. Configure wpsadmin to administer the Domino server. Click Next.
12. Use the values shown in Table 1-1 to modify the next window as needed.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ObjectClass</td>
<td>inetOrg Person</td>
</tr>
<tr>
<td>User DN prefix</td>
<td>cn</td>
</tr>
<tr>
<td>User DN suffix</td>
<td>o=&lt;your domain&gt;</td>
</tr>
<tr>
<td>Group Object Class</td>
<td>groupOfNames</td>
</tr>
<tr>
<td>Group Member</td>
<td>member</td>
</tr>
<tr>
<td>Group DN prefix</td>
<td>cn</td>
</tr>
<tr>
<td>Group DN suffix</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>Administrator DN</td>
<td>cn=wpsadmin,o=&lt;your domain&gt;</td>
</tr>
<tr>
<td>Administrative group DN</td>
<td>cn= wpsadmins</td>
</tr>
</tbody>
</table>
13. Note that the group setting is for *wpsadmins*, and not for the user *wpsadmin*. See Figure 1-91. Click **Next**.

14. Choose **DB2 Universal Database Server** as the back-end database, **Create and Initialize a new Database(DB2 only)** for the Portal Server Database Configuration options, and **Share the Database** for the Do you want to share the database with Member Services option. This is shown in Figure 5-18 in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883. Click **Next** to proceed.

15. Enter *db2admin* as the database user with a password of *db2admin*.

16. This is depicted in Figure 5-19 in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883. Click **Next**.
17. Select **Initialize an existing database** as shown in Figure 5-20 in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883. Click **Next**.

18. Select **Local License Server** as shown in Figure 5-21 in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883. Click **Next**.

19. You will now see a window similar to Figure 1-92.

![checking previous installations](image)

*Figure 1-92  Checking previous installations*

20. Verify that Domino Application Server is running by clicking **Start -> Settings -> Control Panel**. Double-click **Administrative Tools** and then double-click **Services**. The Lotus Domino Server (LotusDominodata) service must be running. If it is not, right-click and select **Start**.

   This is necessary for WebSphere Portal to access LDAP. If it is not running, a window will appear that says **Check if your LDAP server is running** when you start the installation. If you see this window, restart Domino and click **OK**.

   Click **Next** and the installation will begin.

21. Part way through the install, you will get a message to configure admin roles as shown in Figure 1-93 on page 85. Follow the instructions in step 6 in 5.2.11, “Installation Procedure” in *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883.
22. After completing the steps and before clicking OK, make sure that you can access the following URL:

http://<yourFullyQualifiedHostName>/wps/portal

You should get a WebSphere Portal window that says Your portal does not have any page groups as shown in Figure 1-94 on page 86.

If you receive any errors, WebSphere Portal was probably not started correctly. You may need to stop and start the WebSphere Portal again. The portlets install will fail if WebSphere Portal is not started.

Click **OK** when this is working correctly.
Portal server will continue to install. It may take over 30 minutes. If the Installing productivity portlets section goes fast, there might be an error. Check the WPO Setup Manager log and look at the output logs.

23. When install is completed, an Installation Complete window will come up as in Figure 1-95. Click OK and then click Finish.

24. You will need to replace the temporary rt.jar file with the original. Stop the WebSphere Admin Server as described in step 3 on page 77. Delete the file WebSphere\AppServer\java\relib\rt.jar. Rename rt.old in the same directory to rt.jar. Restart the WebSphere Admin Server.
1.3.12 Verify the WebSphere Portal install

Verify the portal installation as described in 5.5.2, “Testing Steps” in IBM WebSphere Portal V4.1 Handbook Volume 1, SG24-6883.

1.3.13 Updating security to enable single sign-on

During the Portal Server install, the WebSphere Application Server Admin ID was switched from dadmin to wpsbind. This was necessary during the install in order for the portlets to be installed correctly. But this configuration may not work for Web Content Publisher and Lotus Workflow.

You will need to perform these steps if:

- You installed Web Content Publisher, and
- Single sign-on fails between Web Content Publisher and WebSphere Portal.

You can verify this by doing the following:

- Log into the URL http://<hostname>/wps/myportal with the username rob and password rob.
- In the same browser session, go to http://<hostname>/wps/wcp.

If you do not receive a prompt to log in again, single sign-on is working properly and you do not have to do the following steps.

Single sign-on is not working

If single sign-on is not working, we need to regenerate the keys that are used for single sign-on in WebSphere Administrator’s Console and then import them into Domino, as follows:

1. Click Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator’s Console. This will open the WebSphere Advanced Administrator’s Console.

2. Click Console -> Security Center. Click the Authentication tab. During installation, WebSphere Portal configured WebSphere Application Server to use the wpsbind account to access LDAP. Since the wpsbind account does not exist within Lotus Workflow, we will use the Domino Administrator (user ID: dadmin) to handle WebSphere Application Server communication with Domino LDAP.

   Modify the fields so they are as follows:
   - Security Server ID: dadmin
   - Security Server Password: (dadmin’s password)
   - Host: <your Domino host name>, such as m23wpn62.itso.ral.ibm.com
   - Directory Type: Domino 5.0
   - Port: <blank>
– Base Distinguished Name: <blank>

Your window should look similar to Figure 1-96.

Figure 1-96 Configure security center to use dadmin user

3. Close the WebSphere Administrator’s Console. Click **Start -> Settings -> Control Panel**. Double-click **Administrator Tools**. Double-click **Services**. Right-click **IBM WS AdminServer 4.0** and select **Stop**. Wait for the service to stop, then right-click **IBM WS AdminServer** and select **Start**.

4. Regenerate the WebSphere Application Server keys as outlined in “Generating keys in WebSphere Application Server” on page 12.

5. Go to the Domino Administrator. Perform steps 1 on page 61, step 2 on page 61 and step 3 on page 61. These steps will start the Domino Administrator and ensure you are logged in with the proper user ID on the proper server.
6. Click **Administration** view and select the **Configuration** tab. Expand **Web -> Web Server Configuration** so the window is similar to Figure 1-97.

![Figure 1-97  Domino Web Server configuration](image)

7. Expand **All Servers** and select the **Web SSO Configuration** document. Click the **Delete** button and a blue garbage can will appear beside it, as shown in Figure 1-98 on page 90. Press the F9 key to refresh and delete the document. This will disable the entry for single sign-on between WebSphere Application Server and Domino.
8. Select the **All Servers** tab, then select **Web -> Create Web SSO Configuration**.

9. Select **Keys -> Import WebSphere LTPA Keys** as shown in Figure 1-99 on page 91.
Chapter 1. Web content management

Figure 1-99 Import WebSphere LTPA Keys into Domino

10. A window will appear requesting the path of the WebSphere LTPA import file. This is located where you saved the DOMWAS.key file in step 4 on page 88. When you have entered the file name, click OK, as shown in Figure 1-100.

Figure 1-100 Enter WebSphere LTPA file location

11. You will now be prompted for the LTPA import file password. Enter it and click OK as shown in Figure 1-101 on page 92.
12. You will see a message that the WebSphere LTPA keys were successfully imported, as shown in Figure 1-102. Click OK.

13. A number of fields will already have been pre-filled from the LTPA file. The LDAP realm will already be specified. Enter the token domain (in this instance, itso.ral.ibm.com) and enter your server name. This is shown in Figure 1-103 on page 93.
Chapter 1. Web content management

14. Click **Save and Close**.

15. Restart the Domino server.

Single sign-on between WebSphere Portal and Web Content Publisher should now be possible. Verify this by using the process described at the beginning of the “Updating security to enable single sign-on” on page 87.

### 1.3.14 Additional configuration for Web Content Publisher

Web Content Publisher comes with an Enterprise Application called WCM Sample that is installed into WebSphere Application Server.

After installing WebSphere Portal on top of Web content management, you cannot preview the WCM Sample project in WebSphere Content Publisher. This is because the context root for the WCM Sample authoring EAR is /WCMSample and the context root for WebSphere Content Publisher is /wps/wcp.
1.3.15 Post-installation

After you have finished installation, you will have noticed several changes to your system. New users and groups have been created, new databases have been created, and new Enterprise Applications have been installed on WebSphere Application Server.

**Web Content Publisher users**

Five users are added during the installation of Web Content Publisher. These users are created as entries in the Domino Name and Address Book and in the WCM database. Each of the user IDs and passwords are the first name of the user, except the WCPAdmin user, which has the password password. WCPAdmin is the administrator of Web Content Publisher.

The created users are as follows:

- WCPAdmin
- Greg ContentContributor
- Dave Developer
- Tara WebMaster
- Rob ProjectLeader

**Tip:** The WCPAdmin user is not configured as an administrator of WebSphere Portal.

**Web Content Publisher groups**

Lotus Workflow creates several groups specifically for Web Content Publisher. These groups are maintained by Domino Directory Services and define the roles that a WebSphere Content Publisher user may or may not perform during the default Lotus Workflow processes. The groups are as follows:

- Content Contributor
- Content Publisher
- Domain Expert
- Workflow Participants
- Project Lead

**Tip:** By default, Web Content Publisher is accessible from http://<hostname>/wps/wcp. The administrator ID is WCPAdmin with an initial password of password.
Web Content Publisher databases

A relational database named WCM is created in DB2 or Oracle. This database is used to store Web Content Publisher information such as user roles, template data, publishing servers, permissions, etc. Structured content is also stored in the database until it is published. Structured content is not stored in the file system.

Additional Notes databases are created in Domino. These databases are used for handling workflow processes in Lotus Workflow. The databases are:

- **LWF Application R3.0.** Used to manage activities and jobs. Monitors current tasks. This database is shown in Figure 1-104 on page 96.
- **LWD Organization R3.0.** Manages the overall organization and participants of Workflow. User workgroups, roles, and departments are managed here. This is shown in Figure 1-105 on page 97.
- **LWF Process Definition R3.0.** Describes the various workflow processes that are created in Lotus Workflow Architect. This database is shown in Figure 1-106 on page 98.
- **LWF Design Repository R3.0.** Used for software reference only, and does not support interactivity through Notes desktop.
Figure 1-104  LWF Application database tab
Figure 1-105  LWF Organization database tab
During installation, additional Enterprise Applications are installed on WebSphere Application Server. Each of these Enterprise Applications are installed as Web modules on WebSphere Portal application server on the host node.

They are as follows:

- WCM
- WCMFR
- WCM Publish WebApp
- WCM Sample
- PersAdmin
- Personalize Email

**WCM**
The WCM Enterprise Application is installed at http://<hostname>/wps/wcp. This is the main engine of Web Content Publisher.
**WCMFR**
The WCMFR application is a default application that serves the file and JSP servlet that accesses files stored in the WebSphere Content Publisher database in order to preview them.

**WCM Publish WebApp**
The WCM Publish WebApp is used to handle the publishing of content from one server to another. This Enterprise Application handles the transfer of data when content needs to be published into a staging or production environment.

**WCM Sample**
WCM Sample is an example project. It serves as an excellent tutorial for administrators of the Web Content Publisher.

**PersAdmin**
This is the application that manages personalization in WebSphere Portal. The Enterprise Application Personalization Runtime is also installed.

**Personalize Email**
E-mail application used with Personalization that supports e-mail-driven campaigns.

### 1.4 Web Content Publisher implementation

This section describes the system administrator’s role in the implementation of Web Content Publisher. It is expected that the reader has read and understood “Web content management fundamentals” on page 2 before continuing.

The system administrator supports Web Content Publisher implementation by:

- Creating Web Content Publisher users
- Managing Lotus Workflow databases, users and groups
- Creating Web Content Publisher Project by:
  - Creating and installing Enterprise Application that displays the Web content
  - Creating database table for structured content
  - Creating a datasource for structured content
  - Creating templates, for authoring, preview, summary and detail
- Creating a publishing server
- Managing versions and editions
This section does not describe in detail the Web Content Publisher application and is not a “how to” guide for the WebSphere Content Publisher Administrator. This information is covered in the Web Content Publisher help, accessible from http://<yourhost>/wps/wcp/helpsystem/en/docFrameset.html.

1.4.1 Creating users

The administrator may be required to create new users for Web Content Publisher. The system maintains Web Content Publisher users in the Domino Name and Address Book and in the WCM database table CMUser. Domino maintains the user’s ID, password, and identification information, and provides this information through LDAP. Domino is also responsible for handling which groups a user belongs to with respect to Lotus Workflow. The WCM database is responsible for managing user permissions with respect to the Web Content Publisher system such as creating new templates, modifying content, etc.

New users to be added both in the Lotus Domino Name and Address Book and explicitly by a Web Content Publisher administrator from the Web Content Publisher Web site.

The process for creating a new user is:

1. Create the user as in steps 21 on page 69 through 24 on page 70, substituting your new user ID for wpsadmin. Click the Register button. This will create your user.

2. From the Administration view, click Groups as shown in Figure 1-105 on page 97.
3. Double-click **Workflow Participants**. Click the **Edit Group** button. Click the **Members** tab. This will bring up a window similar to Figure 1-108. Select the appropriate user and click **Add**. Click **OK**.
4. Click **File -> Database -> Open**. Ensure that the Server field is set to your host name and not Local. Select the **LWF Organization R3.0** database and click **Open**. You should see a window similar to Figure 1-109.

![LWF Organization R3.0 database](image)

**Figure 1-109  LWF Organization R3.0 database**

5. Double-click the **Workflow Participants** workgroup. Click **Edit Document**. Expand **Members**. Your window will be similar to Figure 1-110 on page 103.
6. Click the **Add** button by the Members pane. You will see a window similar to Figure 1-108 on page 101. Select the appropriate users, and click the **Add** button. Click **OK**.

7. Click the **Close** button. You will see a window asking if you want to save your changes. Click **Yes**.

8. You have now created a new user and added the user to the Workflow Participants group in the Name and Address Book and added it to the Workflow Participants group in LWF Organization R3.0.nsf. Restart WebSphere Application Server.

9. Log into Web Content Publisher at http://<hostname>/wps/wcp. Log in as an ID with Web Content Publisher administrative capabilities. The WCPAdmin user has this capability.

10. Click the **Administration** tab on the top right of the window. In the left pane, click **Users** and you should see a window similar to Figure 1-111 on page 104.
11. In the right pane, click the **Add User** icon.

12. Enter the user’s ID into the Add User window and click **Add**. This is shown in Figure 1-112.

**Figure 1-112   Adding a user to Web Content Publisher**

**Tip:** The user will not be allowed access to Web Content Publisher simply by adding a new user to the Name and Address Book and to the Workflow Participants group.
Additionally, you may allow the new user to participate in workflow tasks. Lotus Workflow provides three default workflows:

- Simple Change Process
- Simpler Change Process
- Simplest Change Process

These workflows allow users belonging to certain groups to contribute content, publish content, and reject content. To allow the new user to participate in one of the default workflows provided, you will have to add the user to the applicable groups in the Domino Name and Address Book:

- Content Contributor
- Content Publisher
- Domain Expert

1.4.2 Creating groups for Lotus Workflow

Lotus Workflow requires that a group is installed in the Domino Name and Address Book and then added to the LWF Organization database.

To add a new group:

1. Click Start -> Programs -> Lotus Applications -> Lotus Domino Administrator. This will start the Domino Administrator client. Log into the client, if necessary, with an administrator account.

2. Click the Administration view and select the Files tab. This is shown in Figure 1-113 on page 106.
3. Double-click the `names.nsf` file. Click **Groups** selection in the left -and navigation pane. This is shown in Figure 1-114 on page 107.
4. Click the **Add Group** button. Enter the Group name, and other information and click the **Save and Close** button.

5. Open the LWF Organization R3.0 database by selecting **File -> Database -> Open**. Click **Actions -> Import groups from Name & Address Book** and select the new group. Then you should be able to see this in LWF Architect.

### 1.4.3 Managing Lotus Workflow

Workflow supports the routing of work tasks based on business rules and a person's functional role in an organization. Web Content Publisher provides workflows through Lotus Workflow, an application that is served by the Domino Application Server. The application utilizes the Domino Name and Address book to store user and group information and is implemented with a set of four Domino databases. For more information, refer to “Web Content Publisher databases” on page 95.
Web Content Publisher comes with three default workflow processes:

- **Simple Change Process**
  
  Request a change with a reviewer, receive feedback if the change is valid, and then approve or reject the change.

- **Simpler Change Process**
  
  Similar to Simple Change Process, but does not require an approval to the suggestion.

- **Simplest Change Process**
  
  No approval is required to make a change.

Additional workflows may be created using the Lotus Workflow Architect client. The client provides a GUI to allow non-technical users to define the workflow. It is expected that a development team with Notes programming experience would provide the implementation.


### 1.4.4 Creating Web Content Publisher project

A publishing environment for a given set of users and content in Web Content Publisher is called a *project*. It contains all images, HTML, JSP, cascading style sheets, workflow tasks, etc. The project is the development environment for publishing a Web site.

A Web site may have multiple projects. An example may be a site that has a separate News and Sports sections that are logically separate from each other because they have different rules for approving content, different authors, etc.

**Tip:** Working on two projects simultaneously by opening multiple browsers on the same machine and selecting different projects to work on will cause failures.

Projects are created from the Web Content Publisher site, through a Web browser. Figure 1-115 on page 109 shows the creation of a new project.
The parameters are as follows:

**Name**  
A unique name to identify the project. This is a required field.

**Description**  
An optional description of the project.

**Context Root**  
The context root of the Web module representing the project on the authoring server. The default for the context root is the project name. This is a required field.

**Root Path**  
The default root path used for project import and export. The system does not use the root path at any time other than import and export.
**Default Process**  The workflow process used when creating a new job and identifying a project.

**References**  Not currently used.

**Nature**  Not currently used.

**Quick Edit**  If Yes is selected, allows users to modify a project's content without requiring a workflow task.

**Lock**  If Yes is selected, this setting prevents more than one user from updating the same file at the same time.

**Version**  If Yes is selected, an entry is created for the project in the version control repository.

The two values of most significance to the administrator are the Context Root and the Root Path.

The Context Root is used to map the content for this project to a URL in WebSphere Application Server. An Enterprise Application must be installed to serve content for a project. The value entered in context root is needed when installing the Enterprise Application.

The Root Path specifies a directory for the importing and exporting of projects from a file system or from a version control system. By default it is set to `c:/wcp/<Project-Name>`.

---

**Note:** Manipulating or modifying content in the project root does not affect the content in the Web Content Publisher system because all content is managed inside the database. Therefore, adding an image in the project root directory on the file system will not automatically be detected by Web Content Publisher. The image will have to be imported manually.

---

**Important:** The currently version of Web Content Publisher does not support the deletion of projects or editions. To remove a project, all references to the project in the database must be removed, as well as any published content and unused publish servers.

Importing and exporting projects allows administrators to create backups and allows the migration of file-based content from one environment to another.

When a project is exported to a file system, the system maintains two files for each structured content item and file-based item. Structured content items are in the Structured Content in the Web Content Publisher interface. File-based items are images, HTML and JSPs that are stored in the Files folder in Web Content...
Publisher. The location of structured content and file-based content is shown in Figure 1-116.

**Tip:** Importing a project does not delete the previous project and add the imported project. Any project files that are not in the imported project are still available. Pre-existing files will be overwritten without warning.

![IBM Web Content Publisher - Microsoft Internet Explorer](image)

**Figure 1-116 Location of structured content and files**

Each item of structured content and file-based content such as images and HTML pages generates two files during export. Each item creates one file that contains the item’s metadata. This is stored in the WCM-Meta directory. Another file that contains the data is stored in either the WCM-RESOURCES directory or the WebApplication directory.
The directory structure is Example 1-1.

**Example 1-1  File structure of a newly created project**

C:<root path of project>  
   WCM-Meta  
      WCM-RESOURCES  
      WebApplication  
   WCM-Resources  
   WebApplication  

The project's metadata is located in the WCM-META directory. The metadata is stored as XML files. The WCM-Meta/WCM-Resources directory contains the metadata for each instance of structured content, such as which project it belongs to. These files have a .wcp.xml file extension. An example is shown in Example 1-2.

**Example 1-2  Example of .wcp.xml file in WCM-Meta/WCM-Resources directory**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wcpsample.YourcoToys resourceId="FT0100">  
   <metaData name="LASTMODIFIED" type="java.lang.Long">1023146602364</metaData>  
   <metaData name="PATH" type="java.lang.String">/</metaData>  
   <metaData name="SHAREDACL" type="java.lang.String">0</metaData>  
   <metaData name="PROJECTID" type="java.lang.String">3</metaData>  
   <metaData name="WORKSPACE" type="java.lang.String">base</metaData>  
</wcpsample.YourcoToys>
```

The meta-data for file-based resources is located in WCM-Meta/WebApplication folder. The file format is identical to the format in Example 1-2.

The data that is associated with the metadata is also exported and imported from the file system. There are two types of data in Web Content Publisher: structured content and file-based content. Structured content are the files that are created from authoring templates, and file-based content are items, such as images, that do not have a defined structure.

Structured content is represented in a .wcp file. The file is an XML file that contains the structured data. It does not contain any presentation information. An example is shown in Example 1-3. These files are stored in the WCM-Resources directory.

**Example 1-3  Example of structured data exported to .wcp file**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wcpsample.YourcoToys>  
   <description>YourcoToys</description>  
   <displayName>YourcoToys</displayName>
</wcpsample.YourcoToys>
```
<properties resourceId="FT0100">
  <property name="STAGE" type="java.lang.String">Future</property>
  <property name="DESCRIPTION" type="java.lang.String">Large play station with many compartments for future trips to Mars. Installs on the ground. Base adapts to unpredictable surface conditions. Ages 4-12. Includes laser tag set.</property>
  <property name="AMT_SOLD" type="java.lang.Integer">34562</property>
  <property name="AMT_OVERSTOCK" type="java.lang.Integer">0</property>
  <property name="RETAILPRICE" type="java.math.BigDecimal">0.00</property>
  <property name="WHOLESALEPRICE" type="java.math.BigDecimal">0.00</property>
  <property name="PRODUCTNUMBER" type="java.lang.String">FT0100</property>
  <property name="IMAGEURL" type="java.lang.String">/wps/WCPSample/toys/marsBase.jpg</property>
  <property name="SITE" type="java.lang.String">Raleigh</property>
  <property name="NAME" type="java.lang.String">Mars Play Station</property>
</properties>

File-based content is stored in WebApplication. These files are imported from the file system into the Web Content Publisher (WCM) database as Binary Large Objects (BLOBs).

Creating Enterprise Application for the project

After a new project is created, a system administrator must create an Enterprise Application on WebSphere Application Server that serves the JSP, templates, and content to Web Content Publisher users. If this is not done, the users will not be able to preview their content.

When a new project is created, the system requires a context root. This context root is used by WebSphere Application Server as the URL to present content for the project.

Creation of the Enterprise Application for serving the files in your project is very simple. There are only two files that must be explicitly created: application.xml and Web.xml. We will utilize the WebSphere Application Assembly Tool to generate these files automatically.

Example 1-4 on page 114 is an example of application.xml. The values for <display-name>, <description>, and <web-uri> will be modified accordingly. The <context-root> will be changed to match the context root specified when the project was created, as covered in 1.4.4, "Creating Web Content Publisher project" on page 108.
Example 1-4  application.xml for project’s Enterprise Application

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE application PUBLIC "-//Sun Microsystems,Inc.//DTD J2EE Application 1.2//EN" "http://java.sun.com/j2ee/dtds/application_1_2.dtd">
<application id="Application_ID">
  <display-name>Sports WCM Project</display-name>
  <description>Sports WCM EAR</description>
  <module id="WebModule_1">
    <Web>
      <Web-uri>sportsSection.war</Web-uri>
      <context-root>/wps/sportsSection</context-root>
    </Web>
  </module>
</application>
```

Example 1-5 shows a sample Web.xml. This file refers to the two servlets that will be used to serve content. The display name and description will be configured by Application Assembly Tool.

Example 1-5  Web.xml for project Web module

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Web-app PUBLIC "-//Sun Microsystems,Inc.//DTD Web Application 2.2//EN" "http://java.sun.com/j2ee/dtds/Web-app_2_2.dtd">
<Web-app id="WebApp_ID">
  <display-name>Sports Section Web Module</display-name>
  <description>This is the war for displaying sports section content.</description>
  <servlet id="Servlet_1">
    <servlet-name>Files</servlet-name>
    <description>Files Servlet</description>
    <servlet-class>com.ibm.wcm.servlets.FileResourceServlet</servlet-class>
  </servlet>
  <servlet id="Servlet_2">
    <servlet-name>Jsps</servlet-name>
    <description>JSP Servlet</description>
    <servlet-class>com.ibm.wcm.jasper.runtime.JspServlet</servlet-class>
  </servlet>
  <servlet-mapping id="ServletMapping_1">
    <servlet-name>Files</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
  <servlet-mapping id="ServletMapping_2">
    <servlet-name>Jsps</servlet-name>
    <url-pattern>*.jsp</url-pattern>
  </servlet-mapping>
</Web-app>
```
1. Click **Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Application Assembly Tool.**

2. You will see a window similar to Figure 1-117. Double-click the **Application** icon.

3. Modify the **Display name** field with an appropriate name. Make sure you retain the `.ear` file extension. Fill in some descriptive text for the description field. This is shown in Figure 1-118 on page 116. Click **Apply.**
4. Right-click **Web Modules** in the left-hand navigation pane. Select **New**. Enter a file name for the .war Web Module we will be creating. The context root must be set to the same value specified in 1.4.4, “Creating Web Content Publisher project” on page 108. The classpath is left empty. Add descriptive titles for Display name and Description. Click **OK** when finished.
5. Expand **Web Modules -> <your Web module name>** and right-click **Web Components**. Select **New**. This is shown in Figure 1-120 on page 118.

Figure 1-119  Creating the Web module
Figure 1-120  Create a new Web component

6. Enter Files as the Component Name and enter an appropriate description for the description field. Select Servlet as the Component Type and enter the class name com.ibm.wcm.servlets.FileResourceServlet. This is shown in Figure 1-121 on page 119.
7. Click **OK**.

8. Right-click **Web Components** and select **New**, as done in step 5 on page 117. Enter JSPs as the Component Name and enter an appropriate description for the description field. Select **Servlet** as the Component type and enter the class name `com.ibm.wcm.jasper.runtime.JspServlet`. This is shown in Figure 1-122 on page 120.
9. Click OK.

10. Right-click Servlet Mapping and select New. Enter *.jsp as the URL pattern and select JSPs for the servlet. This is shown in Figure 1-123 on page 121.
11. Click OK. You have now mapped any *.jsp to be handled by the JSPs servlet.
12. Right-click Servlet Mapping and select New. Enter / as the URL pattern and select Files for the servlet. This is shown in Figure 1-123.
13. Click **OK**. You have now mapped URLs ending in / to be handled by the Files servlet.

14. Select **File -> Save**. You will see a window similar to Figure 1-125 on page 123.
15. Enter a file name and click **Save**.

We have successfully created the Enterprise Application file for the Web Content Publisher project. We will now install it on WebSphere Application Server.

**Note:** The .ear file created can be extracted using WinZip. Application Assembly Tool has automatically created the application.xml and Web.xml files. The application.xml file is stored in the /meta-inf directory, while Web.xml is stored inside the .war file, which can also be extracted by WinZip. The Web.xml file is in the /Web-inf directory in the .war file.

16. Click **Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console**. A password prompt may be requested. Enter the appropriate user name and password. By default, this is user *dadmin* with password *password*.

17. Expand **WebSphere Administrative Domain**. Right-click **Enterprise Applications** and select **Install Enterprise Application**.

18. Enter the node that you will install on, and select **Install Application**. Enter the path of the .ear file created in step 14 on page 122. Enter an application name. Your window should look similar to Figure 1-126 on page 124. Click **Next** to continue.
Figure 1-126  Specifying the location of the .ear file when installing the enterprise app

19. You will see the Mapping Users to Roles window. Accept the default and click Next.

20. You will see the Mapping EJB RunAs Roles to Users window. Accept the default and click Next.

21. You will see the Binding Enterprise Beans to JNDI Names window. Accept the default and click Next.

22. You will see a window mapping EJB References to Enterprise Beans. Accept the default and click Next.

23. You will see a window for Mapping Resource References to Resources. Accept the default and click Next.

24. You will see a window to Specify Default Datasource for EJB Modules. Accept the default and click Next.

25. You will see a window for Specifying Data Sources for Individual CMP beans. Accept the default and click Next.

26. You will see a window for selecting virtual hosts for Web modules. Accept the default_host as the default and click Next.
27. You will see a window to select an Application Server for your Web module. This is shown in Figure 1-127.

![Select Application Server for Web module](image)

**Figure 1-127  Select Application Server for Web module**

28. Click the **Select Server...** button. You will see a window similar to Figure 1-128 on page 126. Select **WebSphere Portal** and click **OK**.
Figure 1-128 Select the WebSphere Portal application server for our Web module

29. Return to the original Install Enterprise Application Wizard window and click Next. You will now see a window similar to Figure 1-129.

Figure 1-129 Complete the installation
30. Click **Finish** to install the Enterprise Application. You should see a window that verifies the installation was complete. Click **OK** to continue.

31. Right-click your host node and select **Regen Webserver Plugin**. This is shown in Figure 1-130. This will regenerate the mapping between IBM HTTP Server and WebSphere Application Server to allow IBM HTTP Server to serve files directly to the Web browser rather than going through WebSphere Application Server servlet.

![Figure 1-130 Regen the Web server plug-in](image)

32. Stop and re-start IBM HTTP Server through the Services window.

33. Expand the **Enterprise Applications** tab. Look for the name of the Enterprise Application that you just installed. Right-click it and select **Start** as shown in Figure 1-131 on page 128.
Figure 1-131  Start the Enterprise Application

34. Expand the tab **Nodes -> <your node> -> Application Servers -> WebSphere Portal**. Click **Installed Web Modules**. You should see the newly installed Web module. Your window should look similar to Figure 1-132 on page 129.
Figure 1-132 Starting the Web module

35. Now test to see that the servlet is running. In a Web browser, go to http://<hostname>/<context-root>. You should see a message that says “File not found: null”. This message indicates that the server is properly handling the request, but there is no content file to serve. This is shown in Figure 1-133 on page 130.
1.4.5 Creating structured content

After a project is created, the users of the system may want to create structured content templates. The structured content templates will contain authoring template, preview template, summary and detail templates to handle the input of content and the presentation of content.

Structured content types in Web Content Publisher are created by:

- Defining a content model
- Creating a database based on the content model
- Creating a datasource to access the database through WebSphere Application Server
- Creating a resource using WebSphere Studio Application Developer wizard

**Note:** It may take some time for changes to take place. Wait several minutes before assuming the system is not working.
Importing the resource

Creating templates for authoring, previewing, editing, summarizing and displaying content (optional)

Defining a content model

A content model defines the fields of a structured content template. For example, a press release template might have input fields for a title, author, topic, and body, while a product template might have input fields for a product number, title, description, and price.

The Web content management team decides what fields to define. They must consider the data fields, such as the article title and body, as well as metadata fields, such as the subject or category. Metadata is important if you are planning on implementing a personalization solution or if you are planning on using a site analysis package to determine what information is of interest to your site visitors.

Personalization solutions use metadata for selecting content to show a site visitor. For example, an application may be written to present all articles with a subject Sports to male users under 40. Your Web team can also program your site’s pages (using JSPs or WebSphere Site Analyzer's Web Tracker technology for HTML pages) to record metadata (and possibly regular data) for analysis of how your content is being used.

The data defined in the content model will be applied to the creation of a database table.

Create database table

A database table must be created to represent the content model. Web Content Publisher stores the structured content in a database. The database must be created manually by a database administrator. The table should match the fields in the content model. For example, character fields must be created as CHAR fields with the appropriate length.

The database will be used to create a resource, using wizards in WebSphere Studio Application Developer. The wizards will create Java classes for reading and writing to the database table.

Create a datasource

To access a database from WebSphere Application Server, a datasource needs to be created, as follows:

1. Click Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console.
2. Expand **WebSphere Administrative Domain -> Resources->JDBC Providers->Pers DB Drivers**

3. Right-click **Data Sources** and select **New** as in Figure 1-134.

![WebSphere Advanced Administrative Console](image)

**Figure 1-134 Creating a new data source**

4. Enter a descriptive name for the **Name** field. The JNDI name should be entered as `jdbc/<some descriptive name for your datasource>`. The `databaseName` must contain the name of your database. In our example, we are accessing the WCMDEMO database. Enter the user and password for the user of the database. In our example we used the `db2admin` user. This is shown in Figure 1-135 on page 133.
Chapter 1. Web content management

5. Click **Test Connection**. You should receive a message that Test Connection ran successfully as in Figure 1-136. Click **OK**.

6. Stop and start the WebSphere Portal Application server by right-clicking **WebSphere Portal** and selecting **Stop**. After it has completed, right-click **WebSphere Portal** and select **Start**.

**Creating a Resource using WebSphere Studio**

Web Content Publisher creates structured content through the use of authoring and generation templates. When structured content is initially created in an
authoring template, it is stored in a database. Java code must be written to store and retrieve data from authoring templates into a database.

Web Content Publisher uses resources in WebSphere Personalization to support the communication with a database. Each resource has one or more fixed attributes defined by the schema for the resource. For example, a user resource would contain a first name, last name, and possibly an address, phone number, and customer number. The schema for Web content might include attributes about the content, such as whether or not it is confidential, or to which users it might apply.

Web Content Publisher utilizes WebSphere Personalization’s resource Java APIs to provide access to the back-end database. These classes can be extended to add personalization rules, but it is outside the scope of this book.

**Note:** Additional information on WebSphere Personalization is available from [http://www-3.ibm.com/software/webservers/personalization/](http://www-3.ibm.com/software/webservers/personalization/).

The simplest way to create the required resource classes and the resource descriptor file is by using the Content and User Personalization wizards in WebSphere Studio Application Developer. The Content wizard creates a resource from a database schema. The User wizard creates a resource using an LDAP or a database schema.

1. From within a WebSphere Studio Application Developer project, select the directory in which you want the resource classes to reside. Click the Content wizard icon. The Welcome page for the wizard is displayed.
2. Click the Logon tab to display the window shown in Figure 1-137 on page 135.
3. Enter the information requested to connect to the database. This should access the database created in “Create database table” on page 131 and the datasource created in “Create a datasource” on page 131. Click Connect.

4. The Tables page is displayed showing the tables in the database which you may access for creating the resource. Select one or more tables. If you select multiple tables, then you must identify which one table is the primary table. The other tables are considered associated tables. The Tables page is shown in Figure 1-138 on page 136.
5. Click the **Columns** tab to display the page shown in Figure 1-139 on page 137. Select the columns you want to include in the resource.
6. Click the **Joins** tab if the resource you are defining is comprised of information from multiple tables.

7. Click the **Mapping** tab if you have a column whose value is one of a limited set of abbreviations or codes and you want to map the values to meaningful words. For example, if a particular column in the database held the integer value of 1, 2, or 3 indicating Yes, No, or Maybe, you could map each integer values to the appropriate word. The words would then appear in the Personalization rule editor rather than the codes.

8. Click the **Finish** tab. The page contains the list of files to be generated; see Figure 1-140 on page 138. Click **Finish** to generate the classes.
Once the resource files have been created, you need to copy them to your portal server. The resource files must be accessible in the classpath of the Personalization engine. It is suggested that you copy the files as follows:

- Copy the class files (including the package directory structure) to `was_root\lib\ext`.
- Copy the resource descriptor file, ".hrf" file (including the package directory structure) to the `was_root\personalization\publishedresources` directory. This step is optional, because the .hrf file will be copying in step 3 on page 139.

If using WebSphere Studio Application Developer, you can export the files directly to the file system.

**Note:** For further information on using the User and Content wizards, see the associated help information in WebSphere Studio Application Developer.
The WebSphere Personalization Resource Console is used to import the resource into personalization:

1. Open the WebSphere Personalization Resource Console and log in as an administrator. The URL to open the resource console looks like: http://hostname/wps/PersAdmin/adminframe.jsp.

2. Click the Resource Hierarchies tab.

3. Click Import to display the page shown in Figure 1-141. Specify the path on the portal server (the machine on which WebSphere Personalization is running) where the resource (.hrf) file resides. Click Import File. You should receive a message indicating the import was successful.

Figure 1-141  Importing a resource into personalization

The resource has now been added to WebSphere Personalization Resource Hierarchy. The WCPAdmin must register the resource with Web Content Publisher to make it available for Web Content Publisher users.
Creating a template

Once a resource has been created and added to a project, a user can add an instance of a structured content type. This is done through the Web browser, as shown in Example 1-142.

Figure 1-142 Creating an instance of structured content

Note: New resource collections must be imported into a specific project before they can be used. Instructions are available at http://<Web Content Publisher hostname>/wps/wcp/helpsystem/en/tasks/tc0workwstruct.html#addrc
When a new instance of a structured content type is added, the data is stored in the WCM database table that was created during installation. This database is used to store the data rather than using a structured file format.

Content templates for adding new structured content, editing structured content, and previewing structured content are created but may be replaced with custom templates. Additional detail templates and summary templates can be added by writing JavaServer Pages.

When an instance of structured content is created, the resulting data is stored in a relational database. The database maintains the instances metadata and content. The instance’s metadata and content can be output to a file if it is exported.

The structured content’s data is not converted into a Web-ready file format such as HTML or WML until it has been generated. The data in the structured content instance is combined with a presentation template that describes how to present the data and outputs HTML, WML, or another Web publishable format.

Note: If no templates are specified for a structured content type, the system is still able to add, edit, and preview content.

All structured content with from the same structured content type are stored in the same database, regardless of which edition or project they are from.

1.4.6 Creating a publishing server

After content has been approved, it is ready to be published. How it is published depends on how you have set up your Web Content Publisher project and the process that the content creation is part of. You can define the processes so that some content is published as soon as it is approved. This is applicable to content such as news articles that have an immediate and short shelf life. There are other types of content that you will want to publish in a more coordinated manner.
These are explicitly published. Administrators can do an explicit publish using Web Content Publisher or set up a scheduled publish. By default, only changed content is published, but administrators also have the option of publishing all content.

Content is published via Publish Servers. The receiving servers must install Enterprise Applications on WebSphere Application Server to manage publishing. Files are sent using a series of HTTP requests to the publish targets. Each target is normally a J2EE servlet, but could be anything that follows the appropriate Publish protocol over HTTP. The target servlet receives all project content including files, structured content, and syndicated content.

Web Content Publisher comes with two sample Enterprise Applications to support publishing. They are WCMPznPublish.ear and WCMPublish.ear. During installation of Web Content Publisher, the system will ask what type of server will be installed as part of Web Content Publisher.

WCMPznPublish.ear is used to publish the HTML, JSPs, and other content as files. Additionally, data from authoring templates will be published into a local or remote database. This database that the system will be published to is based on the database that the resource collection is modelled after, as discussed in “Create database table” on page 131.

The advantages of publishing authored data to a relational database is that applications may query the database for specialized results. For example, an application may display all content that is targeted for users over the age of 60.

WCMPublish.ear publishes the content as files. The WCM database tables are not transferred over.

**Note:** Imported HTML files may not properly resolve all of their hyperlinks and not appear correctly in preview mode.

At publish time, files are moved from the transferring database to the new servers file system and database (optional). The publisher specifies which servers they wish to publish, and whether they want all content published, or only the files that have been modified since the previous publish. They can also publish at a specific time. The interface for publishing content is shown in Figure 1-143 on page 143.
Before content can be published, a publish server must be defined, as shown in Figure 1-144 on page 144. Creating publish servers requires only a server name, the servlet URL that manages the transfer of files, any additional proxy settings, and any user ID and password protection that is required to transfer content.
At publish time, the structured content instances that are stored in database tables are aggregated with generation templates to produce files that can be served from a Web server, such as HTML files.

Files are transferred to the receiving machines through a series of requests to the servlet URL specified in Figure 1-144. The servlet takes the files and publishes to the target server’s publish target.

**Tip:** Adding a publish server as shown in Figure 1-144 assumes that the receiving server has already installed the Enterprise Application so that it can act as a publish target. In the example, the WCMPznPublish servlet has been installed on the m23wpn62.itso.rai.ibm.com machine.
The target server receives files based on the configuration of the Enterprise Application. The Web.xml file for the WCMPublish Web-module is shown in Example 1-6.

**Example 1-6  Web.xml for WCMPublish.ear**

```xml
<Web-app id="WebApp_1">
  <display-name>WCM Publish Web App</display-name>
  <description>WCM Publish Web App</description>
  <servlet id="Servlet_1">
    <servlet-name>Publish</servlet-name>
    <description>Publish Target</description>
    <servlet-class>com.ibm.wcm.servlets.PublishServlet</servlet-class>
    <init-param id="InitParam_1">
      <param-name>baseDir</param-name>
      <param-value>washomedir/installedApps</param-value>
    </init-param>
    <init-param id="InitParam_2">
      <param-name>defaultWebAppDir</param-name>
      <param-value>WCMPublish.ear/WCMPublish.war</param-value>
    </init-param>
  </servlet>
  <servlet-mapping id="ServletMapping_1">
    <servlet-name>Publish</servlet-name>
    <url-pattern>/publishtarget</url-pattern>
  </servlet-mapping>
</Web-app>
```

Note that the "washomedir" specified for the baseDir parameter must be changed to a fully qualified directory. The baseDir and defaultWebAppDir are used together as a root directory on which to place the content sent from Web Content Publisher. Using the Web module's context name and the url-pattern shown in the servlet-mapping above, the fully qualified URL for this target is http://<target.host.name>/WCMPublish/publishtarget. This sample target displays the following message if invoked from a browser: Get request not allowed for this servlet. This is a good way to tell if the servlet is set up and configured properly.

If you are using the WCMPznPublish servlet, then data created from authoring templates are transferred to the database, as well as the files.

Security is managed by entering the user name and password according to the WebSphere Application Server security settings on the transferring machine. This restricts transferring servlet to access the servlet on the receiving target's machine.
1.4.7 Managing versions and editions

When content is completed, it can be archived or editioned. Archiving and editions create copies of the project. File resources, such as images, will be duplicated in the database rather than maintaining a reference.

**Tip:** Creating many editions and archives of a copy can result in a large amount of redundant data. Consider the storage impacts when creating archives and editions.

If WebSphere Studio Application Developer installed and working with CVS, you can import and export resources through Web Content Publisher. Information on installing and configuring CVS with WebSphere Studio Application Developer is located in the Web Content Publisher installation guide.
Collaboration

This chapter introduces the Lotus Collaborative Places and Components available with WebSphere Portal Extend. The chapter provides an overview of collaboration and introduces the approaches to setting up WebSphere Portal collaboration. A list of useful references is provided at the end of the chapter.
2.1 An overview

WebSphere Portal supports team coordination through collaboration. Collaboration involves uni-directional or bi-directional interaction among the users of a solution. The following are the types of interactions in a collaborative solution:

- Asynchronous, for example, e-mail
- Interactive, for example, instant messaging
- Broadcast and multicast, for example, video conferencing and team rooms

**Note:** More information on collaboration and other business patterns can be found in *Patterns for e-business*, by Jonathan Adams et al.

WebSphere Portal supports these collaboration models by integrating with such Lotus products as Domino, QuickPlace, Sametime, and Discovery Server.

2.1.1 Collaborative Components

The Collaborative Components allow developers who are writing portlets for WebSphere Portal Server to easily add Lotus Collaborative functionality to their portlets. The Collaborative Components provide the data from collaborative systems to allow the developer to execute actions on the Lotus Collaborative products, while leaving the user interface up to the developer.

The Collaborative Components hide the configuration details of the Lotus products that are installed within the enterprise. Developers using these components can add collaborative functionality to a portlet without regard to server configuration specifics. For example, a developer can use the people awareness tags without having to know the name of the Sametime or LDAP server. The Collaborative Components are implemented in Java and include no platform-specific code. They can be used on any J2EE-compliant server.

**Types of Collaborative Components**

The Collaborative Components fall into two main categories:

- **Java classes and methods (cs.jar)**
  
  This package contains all the Java implementations of the Collaborative Components. There are classes and methods for leveraging Domino, QuickPlace, Sametime, and Discovery Server.

- **JavaScript tag libraries (people.tld and menu.tld)**

  These tag libraries provide Sametime awareness and continual menus to JSPs.
When to use the Collaborative Components

The goal of the Collaborative Components is to expose the most commonly used aspects of the Lotus Collaborative technologies through a simple and consistent API. The components are not a replacement of the core product APIs, but rather are complementary. Developers may choose to use the Collaborative Components when they need quick and easy access to Lotus technologies, and may also use the core product APIs in other portions of their applications when more advanced integration with the Lotus Collaborative technologies is required.

2.1.2 Collaboration portlets

The standard collaboration portlets that are a part of WebSphere Portal Extend include Lotus Notes e-mail, calendar, and to-do list portlets, plus Lotus Notes discussion, document library, and team room portlets. Table 2-1 describes each portlet.

<table>
<thead>
<tr>
<th>Collaboration portlets</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>My iNotes</td>
<td>Provides access to a Lotus iNotes server for Welcome, Mail, Calendar, To Do List, Contacts, and Notebook functions.</td>
</tr>
<tr>
<td>My Notes Calendar</td>
<td>Displays the user's calendar from their mail database. Users may choose to view 1, 2, 7, 14, or 31 days.</td>
</tr>
<tr>
<td>My Notes Mail</td>
<td>Displays the user's inbox from their mail database.</td>
</tr>
<tr>
<td>My Notes To Do</td>
<td>Displays the user's To Do list from their mail database.</td>
</tr>
<tr>
<td>Notes Discussion</td>
<td>Views Notes databases built with the Discussion Database Template.</td>
</tr>
<tr>
<td>Notes Mail</td>
<td>Views a user's inbox.</td>
</tr>
<tr>
<td>Notes View</td>
<td>Views Notes databases.</td>
</tr>
<tr>
<td>Lotus QuickPlace</td>
<td>Displays a Lotus QuickPlace view inside the portlet.</td>
</tr>
<tr>
<td>Sametime Chat</td>
<td>Displays a Sametime chat window inside the portlet.</td>
</tr>
<tr>
<td>Team Room</td>
<td>Views Notes databases built with the Team Room Database Template.</td>
</tr>
</tbody>
</table>

The portlet catalog is frequently updated and can be accessed from:

These portlets can be deployed to leverage Portal collaboration without the need to write custom applications.

2.2 Installing and configuring Portal collaboration

The Redpaper, WebSphere Portal Collaborative Components, REDP0319, provides details for configuring collaboration products and services. We recommend following the instructions in this Redpaper for installing collaboration products. The paper can be downloaded from the IBM Redbooks Web site:

http://www.redbooks.ibm.com

The remaining sections in this chapter focus on considerations while installing collaboration products using the Portal Setup Manager. The concluding section provides additional reference materials that might be useful if you do not wish to use the Setup Manager to actually install the Lotus products.

The Setup Manager for WebSphere Portal (Extend) allows you to install Lotus Collaborative products in addition to the Collaborative Places and Components. In a single-tier install, the Setup Manager would configure both the Portal and the Lotus product for collaboration.

However, a single-tier install for these products is highly unlikely in a production environment. In such cases, the products would need to be manually configured for collaboration. The required settings would vary depending upon the order in which the products are installed.

Generally, a production install for Portal collaboration would be similar to Figure 2-1.

![Diagram of collaboration environment](image-url)
The Lotus Domino stand-alone can be eliminated by installing Sametime or QuickPlace as an overlay on Domino.

2.2.1 Installing and configuring Sametime using Setup Manager

The Sametime.ini file has to be updated to allow the WebSphere Portal to access Sametime services. This file is automatically updated in a single-tier install. However, you would need to update this file in a multi-tier install.

Sametime installed before WebSphere Portal

If you installed Sametime before installing WebSphere Portal, you would only need to update the file, `<SAMETIME_DIR>\Sametime.ini`, after you complete the Portal installation.

In a test or debug environment, you might update the file with the lines shown in Figure 2-2.

```
[Debug]
VPS_BYPASS_TRUSTED_IPS=1
```

*Figure 2-2  Sametime.ini debug settings*

However, in a production environment, you should remove the debug setting specified above and include the following lines in the INI file.

```
[Config]
VPS_TRUSTED_IPS= PortalIP
```

*Figure 2-3  Sametime.ini production settings*

Sametime installed after WebSphere Portal

In this case too, you would need to update the Sametime.ini file as above. However, in addition to that, you would need to update the CSEnvironment.properties file and also create a hostAddress.xml file for your Sametime server.

Details on performing this activity can be obtained from the IBM Redpaper, *WebSphere Portal Collaborative Components*, REDP0319.

2.2.2 Installing and configuring QuickPlace using Setup Manager

In this section, we discuss the activity before and after the QuickPlace install.
QuickPlace installed before WebSphere Portal
This scenario would not require you to take any additional steps. The Portal Setup Manager would update the CSEnvironment.properties file for QuickPlace integration when the Portal is installed.

QuickPlace installed after WebSphere Portal
You would need to update the CSEnvironment.properties file to enable QuickPlace services and update the host name for the QuickPlace server.

2.2.3 More information
The Lotus Developer Domain (http://www-10.lotus.com/ldd/) provides HTML and PDF versions of product documentation and support material. See “Related publications” on page 267 for URL links and additional documentation in PDF format regarding the Lotus products mentioned in this chapter.
This chapter introduces the search capabilities available in WebSphere Portal offerings, specifically portal search and extended search.
3.1 Introduction

Search capabilities form an integral part of a Web portal. The ability to find relevant documents based on a set of keywords is a lifeline for an information portal.

Most portals deploy intelligent and heuristic search engines that work on search indexes spanning millions of Web pages. These indexes can be comprehensive or may be updated based on popular searches. Some sites also provide speciality searches, which essentially means that the search engine searches through an index that points to documents pertaining to a specific domain of interest.

WebSphere Portal provides integrated text search capabilities, including a search portlet, a crawler, and a document indexer. The search service can search the portal's document repository as well as Internet content. WebSphere Portal's built-in search engine is optimized for full-text searching of small and medium-sized collections where precision is essential. It efficiently applies state-of-the-art search algorithms producing high-quality search results.

The search engine supports free-text queries, with query assistance and query word completion. Search queries use advanced query operators (+ or -) to indicate keywords that must be in the document or keywords that must not be in the document. The search engine can search documents in any language and supports synonyms and stop word lists. Search results include document summarization and search results clustering.

The search engine integrated into the Portal is Juru, found at: (http://www.haifa.il.ibm.com/km/ir/juru/).

3.2 Using the integrated document search

Setting up document search for your Portal would require:

1. Creating the Search page
2. Building an index
3. Setting up security
4. Configuring the crawler.properties (optional).

3.2.1 Creating the Search page

You will need to create a page that will contain the Document Search and Manage Search Index portlets. Let us create a sample search page.
1. Log onto the portal as the Administrator (wpsadmin).

2. First, we need to create a copy of the Document Search portlet, which we can then use on our Search page. Select **Portal Administration -> Portlets -> Manage Portlets**.

   **Note:** It is recommended that you create another instance of the Document Search portlet, because this portlet can be used to search on a single index.

3. From the list of portlets, select **Document Search** and then click **Copy**. See Figure 3-1.

![Figure 3-1](https://example.com/figure31.png)

**Figure 3-1**  Create a copy of the Document Search Portlet

4. Provide a name for the new portlet instance, for example, “My Document Search” and then click **OK**.

Chapter 3. Search capabilities  155
5. The new portlet is not activated by default. So, select it from the list of portlets and then click **Activate/Deactivate**.

6. Click **Modify parameters**. This option allows you to specify the search index. Specify the IndexLocation parameter, for example, `/var/PortalServer/indices/index1` or `C:\temp\index1`, depending upon the platform on which the Portal is installed. This is the name and location of the index that we will create later on. Now, click **Save**.

   **Note:** The path `/var/PortalServer/indices/index1` is the location that we have chosen to store our index in. It is not a default setting. Also, multiple indexes cannot share a common location (directory).

7. Select the **Work with Pages** option. Click **Manage Places and Pages** and then select **Create place**.

8. Provide a place name and default locale title for the place, for example, “Test”. Then, click **OK**.

9. From the list of places you can manage, select **Test** and then click **Manage pages**.

10. Click **Create page -> Create new**.

11. Provide a name for the page (for example, “Search”), select **Layout** and then click **OK**.

12. Select **Edit Layout and Content**. For the Place, select **Test** and for the Page, select **Search**.

13. Click **Get portlets**. Select either **Show all portlets** or **Search for portlets** using the keyword “search”. Click **Go**.

14. From the list of portlets returned, select **My Document Search and Manage Search Index** portlets by clicking the add to list (+) button besides them. Then, click **OK**.

15. You can edit the layout of the Search page and then add the selected portlets to the page. Click **Activate**.

### 3.2.2 Building the index

The Manage Search Index portlet can be used to build and maintain indexes of Web content that will be used by the search portlet. The search index stores keywords and terms and maps them to their source documents, enabling fast processing of requests from the search portlet. During the build process, documents are retrieved for indexing through a Web crawler (robot). Searchable resources can be stored on the local portal server or on remote sites. Users can search HTML and text documents.
1. Log onto the portal as the Administrator (wpsadmin) and then navigate to the search page that we created; for example, click Test -> Search.

2. On the Manage Search Index portlet, click Configure search index.

3. Specify the following values for configuring our index (see Figure 3-2 on page 158):
   - Set the location of the index as /var/PortalServer/indices/index1
   - Set the task for configuring the index as New Index
   - Choose the URL as http://www.ibm.com/us/ or any URL that would be the base URL for your index.
   - The Enable CJK language support option enables support for Chinese, Japanese, and Korean languages. We do not require this option.
   - Set the document types to be indexed as both HTML and text.
   - Set the levels of linked documents to at least 1.
   - Retain the number of linked documents to index default of 100.

Note: An existing index can be reconfigured at any time by choosing the Update Index option in the Configure search index window. However, the index has to be rebuilt using the Manage search index option.

Note: If you want to index documents on the other side of an Intranet firewall, you must change the crawler.properties file with the name and port number of the SOCKS or proxy server. Also, you can have a single index for multiple sites. See 3.2.4, “Configuring crawler.properties” on page 162.
Click **OK** to save the configuration and then click **Done**.

4. Now click the **Manage search index** option on the Manage Search Index portlet.

5. From the list of indexes, select the index that we just configured (/var/PortalServer/indices/index1) and then click **Begin index update**.
Once the index has been built, if you re-visit the Manage search index window (or click Refresh on the browser) you will see the statistics for Last update completed at and Number of active documents updated.

6. Click Done.

### 3.2.3 Setting up permissions

There are two basic tasks that are required to be completed before the Search feature can be made available to a portal user:

- Portal users should be provided View access to the Search page.
- The Manage Search Index portlet should not be accessible to users other than the Administrator.

**Note:** The Manage Search Index portlet can be removed from the Search page once the index or indexes have been created. However, you might want to keep it on the page for future administrative tasks.

The following are the steps to accomplish these objectives for our Search page:

1. Log onto the portal as the Administrator (wpsadmin) and then click Portal Administration -> Security.
2. For the Select a group or user to assign permissions field, select Special groups -> All authenticated users.
3. In the Select the objects for the permissions field, select pages. Click Go. See Figure 3-4 on page 160.
4. Provide View permissions for the Test place and Search page. Click **Save**.

5. Now, in the Select the objects for the permissions field, select **portlet**s. Select **Search On**: **Name contains** and enter **search** as the keyword for the field. Then click **Go**.

---

**Figure 3-4** Set View permission for the Search page
Chapter 3. Search capabilities

6. Provide View access for the My Document Search portlet and None for Manage Search Index. Click **Save**.

7. You can now log out and then log onto the portal as an ordinary user. The Search page would look as shown in Figure 3-6 on page 162.
3.2.4 Configuring crawler.properties

The index build process is optimized for crawling inside an Intranet. If you need the crawler to fetch documents on the other side of a firewall, you need to update the crawler.properties file (located in the index directory). You can set either the name and port of a proxy server or a SOCKS server. See Example 3-1.

Example 3-1 Proxy settings for the crawler

```
#The name of the socks server to be used <server name>: 
#<port number>server-name>: 
SocksServer=socks.yourco.domain:1080

#The name of the proxy server to be used <server name>:<port number>
ProxyServer=proxy.yourco.domain:80
```
You can specify additional URLs (maximum of nine) to be crawled into the same index.

```plaintext
#OtherRoot1=http:\//www.second.site
#OtherRoot2=http:\//www.third.site
...
#OtherRoot9=http:\//www.last.site
```

Figure 3-7  Additional sites to be indexed

### 3.3 Federated search

Portlets using IBM Lotus Domino Extended Search R3.7 and Enterprise Information Portal search can access and aggregate other search engines and indexes in a distributed fashion. Customers seeking support for large document collections or for searching a wide range of document types and data sources should consider using IBM Lotus Extended Search or Enterprise Information Portal.

#### 3.3.1 IBM Lotus Domino Extended Search R3.7

Lotus Domino Extended Search provides distributed, heterogeneous searching across Domino servers, databases, and the Internet, without the user having to know the details of these various systems. The result is a single-point of access to a variety of data sources without requiring a new, central index.

Domino Extended Search can search and retrieve documents from repositories that include Lotus Notes 4.X and 5.X, Domino.doc, and R5 Domain Index. It also searches external sources such as Microsoft Index Server and Site Server, LDAP-compliant directories, 18 popular Web search sites and News sites, commercial content providers, and ODBC-compliant relational databases such as IBM DB2, Oracle, Sybase, and Microsoft's SQL Server. Results can be ranked by relevancy over multiple data stores.
3.3.2 Enterprise Information Portal (EIP)

Enterprise Information Portal (WebSphere Portal Experience only) can manage data access across multiple sources such as content management repositories, e-mail systems, relational databases, file systems, Web sites (both intranet and Internet), and more. The Enterprise Information Portal integrates data sources across the enterprise with a unified set of APIs to simplify programming and speed development and deployment, while providing an interface layer that isolates portal applications from changes to underlying data repositories.

Documents can be full-text indexed/searched using the EIP crawler and text search features. Formatted document types handled by IntraNet Solutions (INSO) technology are supported, in addition to standard markup text such as HTML and XML. Documents can be categorized, enabling searching by category. APIs are provided for capturing and storing other metadata about documents.

EIP provides connectors for a variety of repositories provided by IBM, Lotus, and other vendors, such as Documentum and Filenet. Federated searches can be applied across multiple repositories and can exercise searching based on metadata, full text, and other specialized search properties, such as Query by Image Content (QBIC). The Text Analysis features of EIP support creating full-text indexes, and subsequent searching across all the text portions of the content sources configured for use in WebSphere Portal. Sources can be accessed for indexing by the Web crawler or by a metadata search. Portlets for accessing EIP advanced and federated search functions are available from the Portlet Catalog.
Portal security

After a conceptional introduction about the Authentication, Authorization and Administration implementation of WebSphere Portal, this chapter provides information about how to use access control and the Credential Vault system.

It also illustrates two scenarios implementing Secure Sockets Layer (SSL) in a WebSphere Portal environment and discusses common setup difficulties.

For additional information on Portal security, you should review the redbook, *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885 available at:

http://www.ibm.com/redbooks
4.1 Authentication, Authorization, Administration (3A)

Authentication, Authorization and Administration of users is included in the WebSphere Portal implementations. It is also capable of delegating parts or all three of these to external products. The external products can be from third-party vendors and it can be more than one product.

The strategic 3A product from IBM is Tivoli Access Manager and therefore it is supported best by WebSphere Portal.

Authentication

The authentication component is responsible for authenticating users at login. That is, it checks whether a user is who he claims to be. Typically, this is done by requesting information from the user about identity and credentials, such as a password to prove that identity. The authentication component checks whether the credentials that a user provided match the assumed identity. If the credentials are verified successfully, the user is logged in and a session is established.

There are different authentication mechanisms. The most important ones from a server perspective are form-based or basic authentication based on user ID and password. SSL/TLS client authentication is based on digital signatures.

By default, WebSphere Portal uses form-based authentication. Form-based authentication means that a user is prompted through an HTML form for the user ID and password for authentication when trying to access the portal. In a database-only installation, WebSphere Portal validates the user against its own database. However, in a default database with LDAP installation (see Figure 4-32 in IBM WebSphere Portal V4.1 Handbook Volume 1, SG24-6883 for more information), the WebSphere Portal requests that the WebSphere Application Server validate the authentication information against a Lightweight Directory Access Protocol (LDAP) user registry.

WebSphere Application Server uses Lightweight Third Party Authentication (LTPA) as the authentication mechanism. A Common Object Request Broker Architecture (CORBA) credential is used to represent authenticated users and their group memberships. When a user tries to access a protected resource, the application server intercepts the request and redirects the request to the login form. This form posts the user ID and password to the portal that requests the application server to authenticate the user. If the user can be authenticated, a valid CORBA credential is created and an LTPA cookie is stored on the user’s machine.
**Single sign-on**

Single sign-on is often used in conjunction with security. It is also a frequent requirement for a portal, especially an Enterprise Portal. Indeed, one of the base requirements of a portal is single sign-on.

With single sign-on (SSO), after a first successful authentication the client will not be asked for further authentication. He is automatically authenticated for the applications participating in the single sign-on domain.

WebSphere Portal uses a double-realm SSO concept (see Figure 4-1 on page 168).

The client-Web App SSO is a well-known concept from other WebSphere products. A flat implementation of such an SSO leads to parallel operating application servers, such as WebSphere Application Server or Domino Application Server, where both can generate and validate unique credential tokens of users. A scenario as shown in Figure 4-1 on page 168 demonstrates the use of an Authentication Proxy prior to accessing applications within an SSO domain. The Authentication Proxy would then pass proper information to the applications of the SSO domain to make them aware which client it is and that the client was successfully authenticated. With WebSphere Application Servers such as WebSphere Portal in that layer, this is usually done by an implementation of the Trust Association Interceptor (TAI). Applications that do not need to know the identity of the client might assume that all requests are correctly authenticated.

The Portal-Back End SSO is conceptionally similar and typical for a portal that acts as an aggregation engine. However, the portal or really the portlets act as the client, usually in commission of the client itself. WebSphere Portal uses the Credential Vault concept to give the portlets the ability to store and retrieve credentials specific to users and applications. Portlets can also leverage ready-to-use or self-made credential object implementations to authenticate the user for the back-end applications.

The double-realm SSO concept illustrates that the Client (shown in Figure 4-1 on page 168) will authenticate only once to the Authentication Proxy or to the Application Server layer. The Portal administrators and the portlet developers must ensure that the client authenticates to the back-end applications as well.

Therefore the client itself does not need to be aware of the existence of the back-end application even if he uses a user identifier and password for it.
Authorization
The authorization component controls access to all sensitive portal resources, for example pages or portlet instances. Actions on particular portal resources should only be possible after receiving authorization from the access control component. WebSphere Portal has a built-in authorization component implementation; its usage is described in 4.2, “Access control for WebSphere Portal resources” on page 168. The authorization functionality can also be externalized.

Administration
Administration usually refers to the organization of authentication and authorization. That can be, for example, the organization of users and their passwords and permissions. But the possibility to organize and administer users in groups and groups again in groups is a part of it.

The physical implementation relies on the LDAP directory structure, which is an open and standardized format of how to access and organize user-related data.

4.2 Access control for WebSphere Portal resources
WebSphere Portal provides fine-grained access control for the resources that it controls, such as portlets, pages and places. Usage of the access control
possibilities can allow complex scenarios for controlling access to resources. Inside WebSphere Portal, the access control function is encapsulated in a separate component and is called upon whenever portal resources need to be accessed for displaying, modifying, or managing them. The portal core code makes sure that a portal user can view a page and the portlets on a page only, if the required permissions have been assigned.

This section focuses on the access control functionality as it is managed by WebSphere Portal itself. There is also the possibility to externalize the management of resources to a third-party external access control software package, such as Tivoli Access Manager or Netegrity Siteminder.

After a short overview of the Access Control List (ACL) portlet, this section describes some of the options for the highly flexible access control administration of WebSphere Portal.

### 4.2.1 The Access Control List administration portlet

To reach the Access Control List administration portlet, do the following:

1. Open a Web browser and go to the login page, for example, `http://fullhostname/wps/myportal`.

   **Note:** The `fullhostname` is the fully qualified host name of the server where WebSphere Portal is installed. It is essential to always use the fully qualified hostname, but in most configuration, WebSphere Portal is able to redirect you automatically from the host name to the fully qualified host name.

2. Log in as a user that has privileges to work with the Access Control List administration portlet, which is by default wpsadmin.

3. Go to the portal administration place by clicking the drop-down menu in the upper-left corner of the default theme.

4. Open the **Security** tab.

5. Select the **Access Control List** portlet to get a window as shown in Figure 4-2 on page 170.
To use the portlet, do the following:

1. Click the **Get groups and users** button (circled in Figure 4-2) to get to a window for selecting specific users or specific groups. Or select **Special**
Groups to set or view settings for all authenticated users or all non-authenticated users.

2. From the Selected users and groups pane, select which type of resource you want to view or edit for the users you selected in Step 1. You may also further qualify the resources that you intend to view or edit. Use the radio buttons below this pane to do so.

3. Click Go to start your survey.

4. The requested resources and their access control permissions for the selected group or user will appear on the right-hand side of the window. If you edit them, click Save to make them persistent.

4.2.2 Users and groups

Typically, a portal operator will separate its users into groups. Separating smaller groups then again from bigger groups will enable sophisticated structuring of the users in the system.

Note: When you are using an LDAP directory as the user database, grouping users will not lead to branches in the LDAP directory. By default, all users to go to the cn=users branch and all groups to the cn=groups branch. The groups will keep the information of these users in the uniqueUsers field. See “LDAP” in IBM WebSphere Portal V4.1 Handbook Volume 1, SG24-6883 for setting up the LDAP structure during install time.

Access permission for resources can be given to both groups and users. If a user is added to a group, it will inherit the group’s permission. That means a user has all the permissions as his group has. If a user is a member of more than one group, it inherits the highest permission for each particular resource.

This is also true for groups, which will also inherit the permissions of the groups they get added to.

Note: Unfortunately, you will not see the inherited permissions of a group in the Access Control List administration portlet in WebSphere Portal Version 4.1.2. If, however, you add a user to this group, the user will show the inherited permissions.

See 2.4, “Users and Groups” in IBM WebSphere Portal V4.1 Handbook Volume 2, SG24-6920 to understand how to create users and groups, how to assign users to groups, and how to assign groups to groups.
Example of users and groups for permission inheritance

The following explains the users in Figure 4-3:

**Mitch**
Is a member of the Pathfinders group and therefore has a superset of the permissions granted for the Trailblazers Group, the Pathfinders group, and the permissions granted for the user Mitch himself.

**Mac**
Is a member of the Adventurers group and therefore has a superset of the permissions granted for the Trailblazers Group, the Adventurers group, and the permissions granted for the user Mac himself.

**Phil**
Is a member of the Trailblazers Group and the Globetrotter Group and therefore has a superset of the permissions granted for the Trailblazers Group, the Globetrotter Group, and the permissions granted for the user Phil himself.

**James**
Is not a member of a group. Therefore, he has only the permissions granted for himself.

**Pathfinders**
Is a group that is a member of the Trailblazers Group. All users of it will inherit a superset of permissions granted.

**Adventurers**
Is a group that is a member of the Trailblazers Group. All users of it will inherit a superset of permissions granted.
4.2.3 Access control rules

WebSphere Portal access control rules are of the form:

<subject> <permissiontype> <object>

Where:

- **subject** is the subject of a rule, which can be either an individual user or a group that usually has individual users as members.

- **permissiontype** is the type of permission, which can be View, Edit, Manage, Delegate, Copy or Create. They are explained in 4.2.4, “Access control permission types” on page 173.

- **object** is the targeting resource. The resource types are explained in 4.2.5, “Access control resources” on page 176.

A concrete example of a valid access control rule would be:

“User:005” “Edit” “Portlet:Mv6 Mail”

If the user with the user ID 005 accesses a page that includes the portlet Mv6 Mail, WebSphere Portal will use this rule to check the user’s permission. The rules are created by the Access Control List portlet or by using the xmlaccess tool, and they are held persistently in the WebSphere Portal database.

4.2.4 Access control permission types

The permission types in WebSphere Portal are:

- **View**
  
  A subject may view a resource in its predefined configuration. For a portlet resource, it means that the user or group will be allowed to access only the view mode of the portlet.

- **Edit**
  
  A subject may change the configuration of a resource. Permission to edit implies the permission to view. Not all resources have the possibility to change the configuration.

  For portlet resources, the possibility to change the configuration means the possibility to change to the edit mode of the portlet. The edit mode is represented by a pencil in the title bar of the portlet (see Figure 4-4 on page 174). To change to this mode, the user clicks this pencil. The programmer of the portlet needs to make the user aware of which mode he is currently working with, if required.
Being in the edit mode, the portlet will be able to write into its current individual portlet setting. That is a persistent data store unique for each user and portlet instance.

![Figure 4-4 Title bar of a portlet with functionality in edit mode](image)

- **Manage**
  
  A subject may install and remove a resource. This permission also implies permissions to edit and view.
  
  For some resources, such as portlets and pages, WebSphere Portal distinguishes between two levels of modifiable settings:
  
  - Settings that affect all users of a portlet, which can only be changed with the permission to manage that portlet.
  - Settings that affect only the current user of a portlet, which can only be changed by that user, but the permission to edit that portlet would be sufficient.

- **Delegate**
  
  This is the permission that is required to be able to change the access control on a resource object. The delegating subject needs to have the permission to delegate to the receiving subject and to delegate for the specific resource. To delegate a permission on a specific resource, the delegating subject needs to have the permission, which is to be delegated.
  
  For example, user A requires Delegate and Edit permission on portlet X, if he wants to give user B edit permission on portlet X. If user A has only view and delegate permission on portlet X, he will not be able to give any user edit permission for this portlet. See Figure 4-12 on page 181.

- **Copy**
  
  A subject may copy a resource instance together with its configuration. The new instance can be configured independently from the old instance. The creator of the copied instance automatically gets manage and delegate permissions. A copy permission differs from the create permission in that a new resource is created from an already existing resource. Copy, therefore, does not imply create permission.

**Note:** The copy permission is used internally. If you are unsure how to use it, work only with the create permission.
Chapter 4. Portal security

Create

A subject may create instances of a specific resource type. The creator of a resource instance automatically gets manage and delegate permissions. Permission to create is not required in order to be able to copy resources if you have the permission to copy.

Using the Access Control List portlet, you can set the create permission for several resources by selecting **Resource type permissions** (see step 2 on page 171).

Figure 4-5 shows a permission table for the user James, who has create permission for the resource objects Pages, Places and Users.

![James (005) permissions for resource types.](image)

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Active</th>
<th>Minimum</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Create</td>
<td>Delegate</td>
<td>Create</td>
<td>Delegate</td>
</tr>
<tr>
<td>Select All</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pages</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Places</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portlets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource collections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User groups</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Users</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>portlet applications</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

*Figure 4-5  Applying the create permission to various resource types*

Moving the authorization, which means the permission check, to an External Authorization Manager might lead to renaming the described permissions. See Table 4-1 on page 176 for a comparison of the naming of permissions in WebSphere Portal and Tivoli Access Manager.
4.2.5 Access control resources

Access control resources are resource objects that assign access control permissions. The access control resources are grouped in access control resource groups. You will find a list of these groups in the drop-down field for number 2 in Figure 4-2 on page 170.

This section gives a short description of all the access control resource groups:

- User groups

**Mitch (bechilly)** permissions for user groups whose name contains *group.

<table>
<thead>
<tr>
<th>Name</th>
<th>Active</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Manage</td>
<td>Delegate</td>
</tr>
<tr>
<td>Select All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All anonymous users</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>All authenticated users</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Trailblazers Group</td>
<td>☑</td>
<td>☐</td>
</tr>
<tr>
<td>testgroup</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Select All</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-6 Define user group permission for user Mickey Mouse

Figure 4-6 shows an example where the user with the user ID bechilly has manage rights for the group Trailblazers Group. However, he does not have delegate permission. Therefore, he will not be allowed to give other users manage permission.

With manage permission for a user group, the subject will have the permission to modify resource permissions for all subjects in this group.

This would be a typical configuration, if Mitch is the Administrator for the users in the Trailblazers only.
Places

Selecting this access control resource group, only the available places are displayed. In fact, both places and pages are displayed, as you can see in Figure 4-7.

User Mitch has no permissions to even view the Mv6 Administration place. However, he has manage and delegate permissions for the place Test. Manage and delegate permissions are automatically assigned to the user that creates the place.

![Mitch (bechilly)](permissions for places modified since 10/1/02)

<table>
<thead>
<tr>
<th>Name</th>
<th>Active</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>View</td>
<td>Edit</td>
</tr>
<tr>
<td>Select All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mv6 Administration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access Control List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Users and Groups</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Test Portlets</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Select All</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 4-7  Set permissions for places and pages*

Pages

Selecting the pages of the access control resource group, Figure 4-7 is displayed in your Access Control List portlet. It shows both the places and the pages that are located in the places.

Here the Mv6 Administration place has the page’s Access Control List and users and groups. The place Test has only one page, the Test Portlets page. For Test portlets, the user has manage and delegate rights. These permissions are automatically assigned to the user that creates the page.

It is not sufficient to give a user permission to pages only. He also requires at least view permission for the place where the page is included. Otherwise, he will not be able to reference the page and therefore he will not be able to use it in any manner.

Portlet applications

*Note: As of Version 4.1.2 the portlet application permission had no influence on its portlets and it was unclear what effect the change of permissions had at all. If you are unsure, do not use this table.*
Portlets

Selecting the portlets access control resource group, Figure 4-8 will be displayed in your Access Control List portlet.

Here the user with the user ID 005 has the permission to see both view and edit modes of the Mv6 Mail portlet. On the Mail portlet, he will only be allowed to access the view mode of the portlet. No permission is assigned for the UserFriendly2 portlet. This means he will not be allowed to add it to one of his pages nor will he even be aware of the existence of this resource.

See 4.2.4, “Access control permission types” on page 173 for more about the various permission types.

James (005)

<table>
<thead>
<tr>
<th>Name</th>
<th>Active</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mv6 Mail</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>UserFriendly2</td>
<td></td>
<td>☑</td>
</tr>
<tr>
<td>WPS Mail</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>World Clock</td>
<td>☑</td>
<td>☑</td>
</tr>
</tbody>
</table>

Figure 4-8  Set permissions for portlets

Resource type permissions

Figure 4-9 on page 179 shows the table of permissions available in WebSphere Portal. By default, all subjects are granted permission to create places and pages. This is required to enable the Work with Pages place, because the portlets there enable users to create new places and pages.

If you give a subject a permission for a portlet that requires one of those Resource Type permissions during runtime, make sure that you grant him the permission at the same time. If a portlet tries to create another portlet and the subject does not have the resource type create permission on portlets, an error will be printed in the portlet and the appropriate log file.
Figure 4-9  Set permissions for the available resource types

- External access control
  Figure 4-10 sets the permission to declare whether an access control decision is made based on the internal access control service or by an external access control system. A user with manage permission has the option of moving resources to and from external control in the Access Control List portlet.

  It basically allows this user to customize whether the subject will see the right arrows in the upper tables or not.

Figure 4-10  Set the permission to externalize access control for resources

- Resource collections
  A directory path or virtual folder under which content documents are stored.

Permissions for the resource collections are used by the Portal Content Organizer portlet. Refer to Chapter 1, “Web content management” on page 1 for more information about Portal Content Organizer and Web Content Publisher.
Portal

With manage permission for this special resource, the user ID can be used to run the xmlaccess tool. The xmlaccess tool is started with the command:

```
xmlaccess <XML file> <userid:password> <portal config URL>
```

Assuming you create a user with the user ID of wpsadm2 and password of secret, and give him manage permission for the Portal resource (see Figure 4-11), you would be able to replace `<userid:password>` with `wpsadm2:secret`.

![vps adm2 (wpsadm2) permissions for portal.](image)

**Figure 4-11** Set manage permission for the special resource portal

For more information about the xmlaccess tool, see the article *Developing an XML request file for XML Access in WebSphere Portal Version 4.1* at:


or the Portal Configuration Interface of the WebSphere Portal InfoCenter at:


**Note:** You will still need to give this user appropriate rights to whatever your XML file is supposed to do!

The permission to the special resource portal will only enable the user access to read the configuration. For example, changing the access permission for a subject of a portlet requires manage permission of this portlet. By not having the portal permission, the subject will be allowed to give any permission to any user of the portlet that he manages.

For example, the user James has view permission to the Access Control List portlet. He has also edit and delegate permission to the Mi6 Mail portlet. So he will be able to give users view or edit permission to this portlet, but not manage permission. See Figure 4-12 on page 181.

If he had the permission for the portal resource, delegate rights would be enough to give any subject any permission to this portlet.
Chapter 4. Portal security

4.2.6 Assigning permissions

Assigning permissions can be a complex task. Even though it looks very simple to give a user view permission to a certain portlet, for example, such permission changes must be done with prerequisite considerations of this resource in mind.

It is, for example, not enough to give a user view permission to the Install portlet to enable him to use it. The following additional steps are required:

- **Create portlet**
  The user will also need to create permissions of the resource type portlets, since the user will obviously create a portlet when installing a new portlet .war file.

- **Manage Portal**
  The user will also need manage permission of the special resource type Portal, since the user will need to update the Portal configuration when installing a new portlet.

---

### Figure 4-12 Example of not having the permission of the special resource portal

The user wps adm2 has manage permission to the special resource portal, so he can change the access permissions independently of his own permissions for all users, including himself. See Figure 4-13. The only prerequisite is that he have delegate permission to the resource.

### Figure 4-13 Example of having the permission of the special resource portal

<table>
<thead>
<tr>
<th>Name</th>
<th><strong>Active</strong></th>
<th><strong>Minimum</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>View</td>
<td>Edit</td>
</tr>
<tr>
<td>Select All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mv6 Mail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Add user to Admin Role

The user will also need to be added to the WebSphere Application Server Admin Role (see “Setup of Admin Role” in 8.2.8, “WebSphere Portal installation process” in IBM WebSphere Portal V4.1 Handbook Volume 1, SG24-6883 for a description of how to do that). In J2EE terminology, installing a portlet application means installing a Web application, and only users who are part of the Admin Role list are allowed to do that in WebSphere Application Server.

Besides the prerequisite issues of resources, such as portlets, security issues also need to be considered. Assigning a permission to a group will implicitly assign the permission to all members of this group. Access permission to a page does not automatically grant access to the portlets on that page. In this case the portlet frame would appear with a message: You are not authorized to access this portlet.

Having a well-elaborated permission tree in place will make it easier to administer the user structure, with Administrators who handle sub-Administrators who handle sub-sub-Administrators, and so on.

4.3 The Credential Vault system of WebSphere Portal

WebSphere offers a Credential Vault as PortletService. The PortletService interface of the Portlet API enables portlets to use pluggable services via dynamic discovery. The Credential Vault is such a system. It provides portlets with a mechanism for mapping from a user identity to a credential, such as a secret. Therefore portlets do not need to store user credentials as part of the user-specific portlet data.

4.3.1 Back-end single sign-on

Especially by using WebSphere Portal as an enterprise portal, WebSphere Portal might often be used as an aggregation and consolidation engine, integrating various enterprise information systems and presenting them through the portal user interface. Due to their design and because of various security aspects, it is often not possible or not reasonable that they relinquish control of their application security, even if they are now accessed through the WebSphere Portal, not directly by the Web browser of the users.

Those back-end systems should therefore still be able to use their own authentication and authorization mechanisms. The users, however, should not be forced to repeatedly authenticate. Permitting the user to authenticate just once is called a single sign-on solution.
Single sign-on from the portal to the back-end applications allows a client, a user with a Web browser, after logging into the portal to access a number of back-end applications through respective portlets without having to authenticate at each of these back-end applications.

Leveraging the WebSphere Portal Credential Vault system, portlets, usually specific to the back-end system, can log into those systems on behalf of the user. See Figure 4-14 for a schematic description of the single sign-on procedure. A user performs a standard login to WebSphere Portal. The portlets will leverage the Credential Vault (CV) through the WebSphere Portal Java APIs to retrieve valid credentials. Using these credentials, the portlet will be able to perform a login at the back-end application.

4.3.2 The Credential Vault segments and slots

The Credential Vault system can store and manage Principals and Credentials for various back-end resources and various users.

- A Principal would usually be a user ID. It is always a unique identifier for the user on that particular back-end system.
- A Credential would usually be a password string that is used by the back-end system to authenticate the Principal.
Vault segments
In the Credential Vault system, the vault is partitioned into vault segments and the vault segments again can have various vault slots. The slots are specific to the back-end application for the shared slots and specific to user and back-end application for slots that are not shared.

The vault segments map onto a specific vault implementation through corresponding vault adapters (see Figure 4-15). By default, the WebSphere Portal internal implementation will be used. It saves its data in the WebSphere Portal database tables. Tivoli Access Manager's repository could be used as an external implementation of the vault.
The Credential Vault system distinguishes between two different types of vault segments:

- **Administrator managed**
  Only Administrators can create credential slots in such a vault segment. Portlets (that is, users) can set and retrieve credentials from a slot in such a segment if they are authorized. They cannot create slots.

- **User managed**
  Portlets, acting on behalf of a portal user, are allowed to create credential slots in this vault segment.

**Note:** Version 4.1.2 of WebSphere Portal cannot have more than one user-managed vault segment. It exists already by default and does not need to be created.

An internal flag marks whether the segment is to be managed by the administrator or by the user.

Examples of administrator-managed vault segments are corporate resources such as Lotus Notes databases or Intranet passwords.

An example of a user-managed vault segment is a personal POP3 mail box of a user.
Use the following instructions to create a Credential Vault segment using the WebSphere Portal Credential Vault portlet:

1. Log in as a Portal Administration user, which is wpsadmin by default.
2. Select the **Portal Administration** place by clicking it the drop-down in the left upper corner of the default theme.
3. Select the **Security** tab.
4. Go to the Credential Vault portlet by opening the **Credential Vault** tab. You will see a window similar to the one shown in Figure 4-16.
5. Select **Add a vault segment**. You will get a window as shown in Figure 4-17 on page 188.
6. In the Add a vault segment window, you will select a vault where you want to add a new segment. In the Vaults drop-down field (No. 1 of Figure 4-17 on page 188), choose the Default vault implementation based on the WebSphere Portal database. This is the only vault available by default, even if using the Tivoli Access Manager vault repository is also supported. The number in
brackets shows how many administrator-managed segments were already defined for this vault.

7. The Resources within selected vault field (No. 2 of Figure 4-17 on page 188) shows a comma separated list of the names of the resources that are located in the vault.

8. In the vault segment name field (No. 3 of Figure 4-17 on page 188), insert a name. You may also optionally insert a name in the vault segment description field (No. 4 of Figure 4-17 on page 188).

9. Click the **OK** image button (No. 5 of Figure 4-17 on page 188).

10. You will be returned to the Credential Vault introductory window (see Figure 4-16 on page 186). A message at the bottom will tell you if the vault segment was successfully added or not.
Use the following instructions to view or delete Credential Vault segments using the WebSphere Portal Credential Vault portlet:

1. Log in as a Portal Administration user, which is by default wpsadmin, with access to the Credential Vault portlet.

2. Go to the Portal Administration place by selected **Portal Administration** in the upper-left corner of the default theme in Figure 4-17.
3. Select the **Security** tab.

4. Go to the Credential Vault portlet by opening the **Credential Vault** tab. You will see a window as shown in Figure 4-16 on page 186.

5. Click **Manage a vault segment**. You will go to a window as shown in Figure 4-17 on page 188.

6. If you want to delete a vault segment, click the appropriate radio button to do so. You will be prompted with a JavaScript pop-up window and asked to confirm.

7. Leave the window by clicking the **Done** image button.

![Figure 4-18 View and delete Credential Vault segments](image)

**Vault segment slots**

Each vault segment can contain one or more Credential Vault slots, which are logical containers where portlets store and retrieve a user's credentials. A Credential Vault slot contains only one credential per user and is the place where the credential secrets are logically located, that is from an API point of view without handling the physical implementation.

From a physical implementation point of view, the credentials of a user are held in a vault, which could be a database table, with the user identifier and the resource name as unique key. See the vault resource as an additional indirection.

A Credential Vault slot is logically linked to a vault resource. This indirection is the linkage between the logical and the physical implementation.
Even if more than one slot can be mapped to a single resource (No. 4 of Figure 4-19 on page 192), this will rarely be used. An exception would be if two different portlets cannot share the logical vault slot with each other, but must share its physical implementation, the vault resource.

The WebSphere Portal Credential Vault distinguishes between three different types of credential slots:

- A system credential slot stores system credentials. These are credentials where the secret is shared among all users and portlets. This type of credential slot is created in the administrator-managed vault segments.

- A shared credential slot stores user credentials that are shared among the user's portlets. That means that the secret is user specific but the same for all portlets of that user. This type of credential slot is created in the administrator-managed vault segments.

- A portlet private credential slot stores user credentials that are not shared among portlets. That means the credential secret is also user specific as well as specific to a concrete portlet instance. This type of credential slot is created in the user-managed vault segment.

Use the following instructions to create a Credential Vault segment slot using the WebSphere Portal Credential Vault portlet:

1. Log in as a Portal Administration user, which is wpsadmin by default.
2. Go to the Portal Administration place by clicking Portal Administration in the upper-left corner of the default theme.
4. Go to the Credential Vault portlet by opening the Credential Vault tab. You will see a window as shown in Figure 4-16 on page 186.
5. Select Add a vault slot. You will see a window as shown in Figure 4-19 on page 192.
6. To select a vault where the segment is located to which you want to add the new vault slot, go to the drop-down list at No. 1 in Figure 4-19 on page 192. The default vault is the one that maps to the default implementation of the vault on the base of the WebSphere Portal database. This is the only vault that is available after a default installation.
7. Insert a unique name for the slot (No. 2 in Figure 4-19 on page 192).
8. Select the vault segment to which you want to add this slot (No. 3 in Figure 4-19 on page 192). The drop-down field lists all available administrator-managed vault segments.
9. In the drop-down list at No. 4 of Figure 4-19 on page 192, you have the choice to create a new vault resource for this slot or to use an already existing resource.

In practice, it is very unusual to have more than one slot pointing to a resource. However, in a rare case it might be required that two different portlets cannot share the logical vault slot with each other, but must share its physical implementation, the vault resource.

**Note:** Be careful pointing more than one slot to a resource, because this might lead to a challenging task for the Security Administrator.

If you are unsure, always create a new vault resource while creating a new vault slot.

10. At No. 5 of Figure 4-19 on page 192, you can check the box to share the slot and therefore the user ID and password for all users.

If you check the box, you will create a system credential slot. You will be able to provide the user ID and password that will be used for all users in the fields below the check box.

If you do not check the box, you will create a shared credential slot. The info fields below will not be enabled, since the user ID and password will not be shared among users.

11. Optionally, add a description in the input field at No. 6 of Figure 4-19 on page 192. Use the link at No. 7 to add the description additionally in one of the various supported languages.

Click the OK image button (No. 5 of Figure 4-19 on page 192) to return to the Credential Vault introductory window (Figure 4-16 on page 186). A message at the bottom will tell you if the vault slot was successfully added or not.
4.3.3 The Credential Vault Service

Note: This section was taken from *Integrating WebSphere Portal Version 4.1 with your security infrastructure*, a whitepaper written by Ingo Schuster, Frank Seliger and Thomas Schaeck. It was added for completeness. The usage of the Credential Vault Service is described in *IBM WebSphere Portal Development Handbook*, SG24-6556 available at [http://www.ibm.com/redbooks](http://www.ibm.com/redbooks).
The Credential Vault Service offers the following functions:

- Map the requested credential slot, the user ID, and the portlet ID to a resource in the vault. A portlet can only retrieve a credential if a respective mapping rule exists. Each credential slot is associated with a certain vault implementation (the actual store). This allows different credentials to be kept in different physical stores.

- Retrieve the user's credential (secret). Some secrets will be stored and managed by the portal (which always uses the local default vault store). If a user secret is not stored in the portal's local vault, it will be acquired from the respective external vault.

- If a credential (secret) is not available, or the authentication fails, an appropriate exception is thrown. The service passes this exception to the portlet, to allow appropriate error handling, for example by asking the user to set the credential through the portlet's edit mode.

- The credential vault will not allow any other person than the credential owner to manage and/or use the credentials – not even the portal administrator. This is done in order to get the necessary acceptance and trust from the end user. A method to access another user's credentials will not be provided.

- There is no general user interface that allows portal end users to manage their credentials in the vaults. With WebSphere Portal 4.1 it is the portlet's responsibility to provide the user in the portlet's edit mode with functions for managing the slots that are used by the portlet.

The portal engine, however, does provide all interfaces required to write a general credential management portlet for portal end users.

- Usually, a portlet “binds” the credentials that it needs to certain credential slots only at runtime, not during deployment.

Portlets that need a credential to complete their service have basically two options:

- Use an existing credential slot that has been defined by the portal administrator in an administrator-managed vault segment.

- Create a new credential slot in the user-managed vault segment.

Portlets obtain credentials by obtaining a CredentialVaultPortletService object and calling its getCredential method. With the returned credential, there are two options:

- Use passwords or keys from a passive credential, passing them in application-specific calls. Portlets that use passive credentials need to extract the secret out of the credential and do all the authentication communication with the back-end application.
Call the authenticate method of an active credential. Active credential objects hide the credential's secret from the portlet, with no way to extract it out of the credential. Active credentials provide additional methods to perform the authentication.

The latter case allows portlets to trigger authentication to remote servers using basic authorization, SSL client authentication, digest authentication, or LTPA without knowing the credential values. Using active credentials means that the portal authenticates on behalf of the portlet, and the portlet can simply use the open connection. While this may not be possible for all cases, it is the preferred technique. For secure transmission of data, portlets can request a secure session (HTTPS) for accessing Web applications.

4.4 Using Secure Sockets Layer (SSL) to access WebSphere Portal

**Important:** Make sure you install FixPack 3a if you intend to use SSL as described here.

WebSphere Portal Version 4.1.2 created some of the links to images and style sheets using a full Uniform Resource Identifier (URI) instead of a server-relative URI. Those elements would still be accessed using HTTP instead of HTTPS, as the schema is hard-coded. Even this would not hurt functionality, depending on the Web browser setup. This would lead to pop-up windows that inform the user about unsecure elements on the page. Those shall be avoided because of security reasons and to avoid unsettling users.

As of Version 4.1.3 of WebSphere Portal, this problem is fixed. In the following setup we used WebSphere Portal 4.1.3a, and we discourage the use of any pervious version in such a setup.

**Note:** Creating an SSL certificate and setting up a WebSphere Application Server are discussed in 4.4.2, “Creating an SSL certificate” on page 196 to 4.4.4, “WebSphere Application Server setup” on page 204. They are also described in the *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176. As the description there targets a Windows environment, we chose here an AIX environment to show the required steps for a successful setup.
4.4.1 Environment topology

The sample setup as described in this chapter looks similar to the one in Figure 4-20.

![Diagram of setup](image)

**Figure 4-20** Deployment of a SSL terminating HTTP Server in the DMZ

We will first create a SSL certificate for the HTTP Server so that WebSphere Portal will be able to serve pages via SSL at all. We will then configure WebSphere Application Server and WebSphere Portal that the public pages are served via HTTP, but the private pages are served via HTTPS (see also conceptional Figure 4-21). Furthermore, we will separate the HTTP Server from the WebSphere Application Server, which produces the setup shown in Figure 4-20.

It would be also possible to have an SSL-secured connection from the HTTP Server to the WebSphere Application Server. See IBM WebSphere V4.0 Advanced Edition Handbook, SG24-6176 for a description of how to set this up. Consider, however, the usage of IPSec between the HTTP Server and the firewall as shown in Figure 4-20. Most operating systems come with this functionality included.

![Diagram of public and private pages](image)

**Figure 4-21** Public and private pages served by HTTP and HTTPS protocols
4.4.2 Creating an SSL certificate

**Important**: In a production environment, you will very likely not create your own self-signed certificate, but buy one from a Trusted Certification Authority such as VeriSign or Thawte.

The IBM HTTP Server comes with an easy-to-use utility, the IBM Key Management, to create self-signed certificates.

To first create a certificate trust database and then a self-signed certificate, complete the following steps:

1. Log in as root user and issue the command:
   
   ```
   # ikeyman
   ```

   In some cases, you might need to set the JAVA_HOME environment variable. For AIX, this would be the command:

   ```
   # export JAVA_HOME=/usr/WebSphere/AppServer/java
   ```

   You will see a graphical user interface of a utility similar to Figure 4-22 on page 197.
2. Select **Key Database File** from the menu bar, then select **New**.

3. In the New window, enter the following and then click **OK**:
   - **Key Database Type:** CMS key database file
   - **File Name:** portalssl.kdb (must be the same as in httpd.conf)
   - **Location:** `<http_server_install_path>/ssl`

4. In the Password Prompt window, as seen in Figure 4-23 on page 198, enter the following, then click **OK** to continue:
   - **Password:**
     - Password to protect keystore file contents
   - **Check Set expiration time**
     - Enter a number of days after which the password will expire. If no expiration is required, uncheck this setting.
– Check **Stash the password to a file?**

**Note:** The IBM HTTP Server accesses the password-protected keystore file `<filename>.kdb` using the password contained in the `<filename>.sth` stashfile. Consequently, the stash option must be enabled.

![Password Prompt](https://example.com/passwordPrompt.png)

**Figure 4-23 Specify password and expiration date of keystore file**

5. Click **OK** when the Information window appears with the message:
   
   The password has been encrypted and saved in the file: 
   /usr/HTTPServer/ssl/portalssl.sth

6. Select **Key Database File** from the main menu, then select **Open**.... Specify the keystore database file and click **OK**. Our example uses `portalssl.kdb` with the path `/usr/HTTPServer/ssl/`.

7. Select **Create** from the menu bar, then select **New Self-Signed Certificate**....

**Note:** If you are enabling SSL for a production environment, select **New Certificate Request** instead. It is strongly recommended that self-signed digital certificates not be used in production.
8. In the Create New Self-Signed Certificate window, shown in Figure 4-24, enter the following values, then click OK:

- Key Label: <user defined label>
- Version: X509 V3
- Key Size: 1024
- Common Name: <hostname.domain.com>
- Organization: IBM
- Organization Unit: ITSO
- Country: US
- Validity Period: 365 Days

![Create New Self-Signed Certificate](image)

Figure 4-24 Specify settings for new self-signed certificate

9. The new certificate should be listed in the Personal Certificates pane.


### 4.4.3 HTTP Server Setup

To enable IBM HTTP Server for using SSL, you have to edit its configuration file httpd.conf, located at /usr/HTTPServer/conf/httpd.conf at an AIX installation.

To do this configuration, complete the following steps:

1. Log in as a root user and stop the IBM HTTP Server by using the command:

   ```bash
   # /usr/HTTPServer/bin/apachectl stop
   ```
2. Back up your current httpd.conf file. For example:

   # cp -p /usr/HTTPServer/conf/httpd.conf /usr/HTTPServer/conf/httpd.nossl

3. Use an editor such as vi to open the httpd.conf file. For example:

   # vi /usr/HTTPServer/conf/httpd.conf

4. Ensure that the following lines are uncommented by removing the # symbol:

   Note: If these lines do not exist, add them below the section of the statements that start sequentially.

   For example AddModule statements that are not below the ClearModuleList statement will not be loaded.

   – LoadModule ibm_ssl_module libexec/mod_ibm_ssl_128.so
     or for Windows systems:
     LoadModule ibm_ssl_module modules/IBMModuleSSL128.dll
   – AddModule mod_ibm_ssl.c
     on UNIX systems only.
   – Listen 80
     Listen 443
   – <VirtualHost hostname.domain.com:443>
     You must substitute your fully qualified host name in this line, which is in our example
     <VirtualHost m10df4ff.itso.ral.ibm.com:443>
     SSLEnable
     </VirtualHost>
     SSLDisable
     Keyfile “/usr/HTTPServer/ssl/portalssl.kdb”
     Make sure, that this path points to the key database file that you created in 4.4.2, “Creating an SSL certificate” on page 196.
     SSLV2Timeout 100
     SSLV3Timeout 1000

5. Ensure the following settings have been removed from the httpd.conf file or disabled by adding the # symbol to the start of each line:

   #AfpaEnable
   #AfpaCache on
   #AfpaLogFile <log_file_path>
6. Save the changes and close the editor.

7. Start the IBM HTTP Server by using the command:

   `# /usr/HTTPServer/bin/apachectl start`

8. Use a Web browser to verify the correct setup of SSL at your IBM HTTP Server. Request the server with an HTTPS schema in front of its fully qualified host name. For our example this would be as follows (see also Figure 4-26 on page 203):

   `https://m10df4ff.itso.ral.ibm.com/`

9. You will be prompted with the certificate that you just created. It is unknown to the browser and therefore it asks the user if it should continue to load data from this site.

   This certificate information window looks different for the various Web browsers. An example is shown in Figure 4-25 on page 202.
10. Every Web browser will indicate somehow that the data it just loaded arrived encrypted (see the arrow in Figure 4-26 on page 203).
Chapter 4. Portal security

11. Since we intend to run WebSphere Portal only partly with SSL, make sure the IBM HTTP Server can still deliver unencrypted pages. Check this by accessing the same URL with an HTTP schema. In our example this would be:

   http://ml0df4ff.itso.ral.ibm.com/
4.4.4 WebSphere Application Server setup

To enable WebSphere Application Server for using SSL, you have to make sure that there exists a host alias that accepts requests on port 443. To do this, complete the following steps:

1. Log in as a root user and start the WebSphere Application Server AdminConsole by using the commands:

   ```bash
   # cd /usr/WebSphere/AppServer/bin
   # ./adminclient.sh
   ```

   **Note:** WebSphere Application Server needs to be up and running to start the Admin Console.

2. Select the **Virtual Hosts** folder.

3. If not already configured, click the **Add** button and add the line:

   ```bash
   *:443
   ```

   to the Host Aliases table as shown in Figure 4-27 on page 205.
4. Click the **Apply** button.
5. Make sure the Default Server is started.
6. Regenerate the Web server Plugin. Right-click the node name and select the option **Regen Web server Plugin** as shown in Figure 4-28 on page 206.
7. Restart the IBM HTTP Server. To do this, use the following command:
   
   `# /usr/HTTPServer/bin/apachectl restart`

8. Close the AdminConsole.

9. Use a Web browser to verify the correct setup of SSL at your WebSphere Application Server. Request the Snoop Servlet with an HTTPS schema. For our example this would be as follows (see also Figure 4-29 on page 207):

   `https://m10df4ff.itso.ral.ibm.com/servlet/snoop`
4.4.5 WebSphere Portal Setup

To enable WebSphere Portal for SSL, some configuration files need to be edited. To do this complete the following steps:

1. Stop WebSphere Application Server.
2. Open the ConfigServices.properties file that is located in the directory `<was_root>/lib/app/config/services/` in an editor.
3. Change the following two properties with the following parameters:
   
   ```
   redirect.login.ssl = true
   host.port.https = 443
   ```
4. Save and close the ConfigServices.properties file.
5. Open the Web.xml file of WebSphere Portal Server that is located in the directory `<wps_root>/app/wps.ear/wps.war/Web-INF/` in an editor.
6. Change the login URL so that it uses an HTTPS schema:

   ```
   <form-login-page>
   https://m10df4ff.itso.ral.ibm.com/wps/portal/.scr/Login
   </form-login-page>
   ```
7. Save and close the Web.xml file.
8. Change links to make them use of HTTPS instead of HTTP.

Edit all JSPs that provide the Login button. In all default HTML themes, the Login button is located in the Banner.jsp file. Make sure you edit every single Banner.jsp file of each theme. If you also take advantage of I-Mode and WML, make sure you edit the appropriate Default.jsp file.

The following files need to be edited after a default installation:

<wp_root>/app/wps.ear/wps.war/themes/html/Banner.jsp
<wp_root>/app/wps.ear/wps.war/themes/html/Corporate/Banner.jsp
<wp_root>/app/wps.ear/wps.war/themes/html/Engineering/Banner.jsp
<wp_root>/app/wps.ear/wps.war/themes/html/Finance/Banner.jsp
<wp_root>/app/wps.ear/wps.war/themes/html/Science/Banner.jsp
<wp_root>/app/wps.ear/wps.war/themes/chtml/Default.jsp
<wp_root>/app/wps.ear/wps.war/themes/wml/Default.jsp

**Note:** If you customized or created your own theme, make sure that you change every tag that leads to the Login page.

Insert the flag `ssl="true"` in all tags that lead to the Login page. The following are examples of the Login tag with the change in **bold**:

- Example of a Login tag in a JSP that creates HTML:

  ```xml
  <wps:if loggedIn="no" notwindow="Login">
  <td valve="middle">
  <a href='"<wps:url ssl="true" home="public" window="Login"/>'><img src='"<wps:urlFindInTheme file="nav_login.gif"'/>" alt='"<wps:text key="link.login" bundle="nls.engine"/>" border="0" align="absmiddle" width="25" height="25" title='"<wps:text key="link.login" bundle="nls.engine"/>""></a>
  </td>
  </wps:if>
  ```

- Example of a Login tag in a JSP that creates Compact-HTML:

  ```xml
  <A href='"<wps:url ssl='true' home='public' window='Login'/>">
  [<IMG src='"<wps:urlFindInTheme file='nav_login.gif'"/>"
  BORDER="1">]
  </A>
  ```

- Example of a Login tag in a JSP that creates WML:

  ```xml
  <go href='"<wps:url ssl='true' home='public' window='Login' /">'/
  ```

9. Change the links to make them use HTTP instead of HTTPS.

You might want to change some more JSPs so that output is now delivered via SSL so that they link back to pages, which is not provided via SSL.

A typical example is the Login.jsp file. The Login page should obviously be provided via SSL to assure the user that his login information will be submitted securely. Just having the POST request for the Login page set to
SSL is discouraged, since users cannot be sure when inserting their user ID and password that this information will be secure.

The Cancel button of the Login page should, however, link back to the WebSphere Portal start page using the HTTP schema instead of HTTPS. Therefore, change the Login.jsp files, located at:

<wp_root>/app/wps.ear/wps.war/windows/html/Login.jsp

Example of a Login tag in a JSP that creates HTML:

```html
<a href="<wps:url ssl='false' home='public' reqid='no'/>
style="text-decoration:none;" title='<wps:text bundle="nls.registration" key="button.cancel" />
alt='<wps:text bundle="nls.registration" key="button.cancel" />
</a>
```

10. Start WebSphere Application Server.

**Note:** It is important that you restart your WebSphere Application Server node that the changes in the Web.xml take effect.

11. Use a Web browser to verify the correct setup of SSL at your WebSphere Portal. Request the Portal Public Page. For our example, this would be:

http://m10df4ff.itso.ral.ibm.com/wps/portal

12. Click the key symbol, which is the Login icon, in the upper-right corner. You will be switched to SSL and the Login page will be delivered using HTTPS (see Figure 4-30 on page 210).

13. Log in and assure that the pages are all delivered via HTTPS.

14. Close the browser and open it again to perform a second test. Request the Portal Customized Page directly. For our example, this would be:

http://m10df4ff.itso.ral.ibm.com/wps/myportal

15. You will get switched to SSL and get the Login page delivered using HTTPS (see Figure 4-30 on page 210).
4.4.6 Forcing usage of SSL

With the setup as explained in the previous sections, users will be able to change the schema in the Login page manually and then transmit their credentials encrypted and also see their authenticated pages unencrypted, in the same way as they would if you had not set up SSL at all.

In most cases, setting up with no enforcement is desirable, because it provides the developers with the best possible flexibility. And also from an administrative point of view, it might be helpful to have HTTP for applications that do not really require SSL, to reduce the load on your servers.

In some setups, it might be desirable to force users to use SSL for their private pages, which can be viewed after authentication. The reasons could be legal-based or based on business rules.

Administrators must decide how to do this enforcement best. Assuming that the SSL is terminated at some incoming Reverse Proxy, WebSphere Portal will not be able to see what protocol the user had in his Web browser. Assuming this is not the case, you could take advantage of the Java Servlet Specification implemented in WebSphere Application Server. To prevent unsecure access to sensitive data, the Java Servlet Specification defines the user-data-constraint element of the Web.xml file, the deployment descriptor for Web applications. For WebSphere Portal, the transport-guarantee field defines the keyword NONE by
default. Change this keyword to CONFIDENTIAL if you want the WebSphere Portal to enforce secure transport. After this change, the WebSphere Portal will refuse all requests to its secure pages (for example \(<\text{hostname}>/wps/myportal\)) that are not requested via SSL. Find a description of these settings in the WebSphere Application Server documentation at http://www-3.ibm.com/software/Webservers/appserv/doc/v40/ae/infocenter/was/0606080004aa.html.

**Note:** We only recommend this change if you fully understand the implication for a J2EE application such as WebSphere Portal.

### 4.5 Using a Remote HTTP Server

A very common topology setup is to have the HTTP Server in a demilitarized zone and the WebSphere Portal behind another Firewall in the back end.

To do this, complete the steps outlined in the following sections.

**Note:** These steps apply to both HTTP and HTTPS.

The steps are done using an AIX platform as an example and assume a setup of the environment as described in 3.4, “WebSphere Portal for AIX prerequisites” in IBM WebSphere Portal V4.1 Handbook, Volume 1, SG24-6883.

To find out about these steps for other platforms, refer to the IBM WebSphere V4.0 Advanced Edition Handbook, SG24-6176.

**Install IBM HTTP Server and the WebSphere Application Server Plugin on the remote machine**

1. Insert WebSphere Portal CD #3-1 (WebSphere Application Server for AIX and Solaris) and issue the following commands as a root user:

   ```
   # mount /cdrom
   # cd /cdrom/was/aix
   # ./install.sh
   ```

2. The Welcome window is displayed. Click **Next** to continue.

3. The Install program will present you with an information window about the required operating system dependencies (see Figure 4-31 on page 212). Click **OK** and continue, if the requirements are fulfilled.
4. On the Installation Options window, select **Custom installation** and click **Next** to continue.

5. On the Choose Application Server Components window, choose **IBM HTTP Server 1.3.19** and **Web server Plugins**, or only Web server Plugins if you already have an appropriate HTTP Server installed (see Figure 4-32 on page 213).
6. In the Choose Application Server Components window, select the type of HTTP Server you want to use. If you selected IBM HTTP Server 1.3.19 in the previous window, make sure you select IBM HTTP Server Plugin now (see Figure 4-33 on page 214).
7. In the Select Destination Directory window, select the destination directory for the WebSphere Application Server Plugin. By default, this is set to /usr/WebSphere/AppServer. Click Next to continue.

8. The Install Options Selected window informs you about the options you selected for installation. Click Install to start the installation.

9. In the Location of Configuration files window, you are asked to insert the path to your HTTP Server configuration file. For an IBM HTTP Server installation, insert the full path to the httpd.conf file, which is /usr/HTTPServer/conf/httpd.conf on a standard AIX installation (see Figure 4-34 on page 215).

Click Next to continue.
10. In the Setup Complete window, click **Finish**.

**Install WebSphere Application Server Fixpack 2**

The Fixpack of WebSphere Application Server just affects the Plugin itself. However, it will also update the level of the IBM HTTP Server as well as add some fixes to the JDK that comes with WebSphere Application Server. Therefore, it is necessary to have the same level of WebSphere Application Server on the remote machine as you have on the WebSphere Application Server machine itself. You will, however, not require any of the e-fixes.

**Note:** Make sure that none of the components you intend to update is running. For example, check for running httpd processes.

1. Insert WebSphere Portal CD #3-1 (WebSphere Application Server for AIX and Solaris) and issue the following commands as a root user:

   ```
   # mount /cdrom
   # cd /cdrom/was/aix
   # ./install.sh
   ```

2. Insert the installation directory of the WebSphere Application Server and press Return. In our example, the WebSphere Application Server directory is:

   `/usr/WebSphere/AppServer`

3. Insert the path where the installer can write temporary files for the WebSphere Application Server Fixpack and press Enter. In our example we use the `/tmp` directory.
4. Insert the path where the installer can write temporary files for the JDK PTF and press Return. In our example we use the /tmp directory.

5. On the question whether you want to the install IBM HTTP Server PTF, type y for yes and confirm by pressing the Enter key.

6. Insert the path where the installer can write temporary files for the IHS PTF and press Return. In our example we use the /tmp directory.

7. On the question whether you want to the install Java2 Connector Architecture Implementation update, type y for yes and confirm by pressing the Enter key.

8. Insert the path where the installer can write temporary files for the J2C PTF and press Return. In our example we use the /tmp directory.

9. Back up and replace your mod_ibm_app_server_http.so file. Issue the following commands as a root user to do this:

   # cd /usr/WebSphere/AppServer/bin
   # cp -p mod_ibm_app_server_http.so mod_ibm_app_server_http.so.bak
   # cp /cdrom/ihs/plugins/aix/mod_ibm_app_server_http.so .

   See also “Installing Cache Plug-In for IBM HTTP Server” in IBM WebSphere Portal V4.1 Handbook, Volume 1, SG24-6883 to see how to install it on other operating systems.

   **Note:** We were not able to get any description of this specific HTTP Server plug-in. We assume it is not possible to configure it. It is intended to give you performance improvements for static content, but does not enhance functionality.

   Especially in cases where you already have a Caching Proxy in place, you might want to decide to skip this and the following step.

10. Insert the WebSphere Portal CD #3-1 (WebSphere Application Server for AIX and Solaris) in the WebSphere PortalServer machine and copy the FileServingServletESI.jar to the classes directory of the WebSphere Application Server. To do this, go to the other machine and issue the following commands as a root user:

    # mount /cdrom
    # cd /cdrom/ihs/plugins/aix
    # cp FileServingServletESI.jar /usr/WebSphere/AppServer/classes/
Configure WebSphere Application Server to use the remote HTTP Server plugin

Some additional configuration steps are required to enable the correct usage of the remote HTTP Server. Therefore, complete the following steps at the machine where WebSphere Portal is installed:

1. Start the WebSphere Application Server Admin Console by issuing the following commands:
   
   ```
   # cd /usr/WebSphere/AppServer/bin
   # ./adminclient.sh
   
   # cd /usr/WebSphere/AppServer/bin
   # ./adminclient.sh
   ```

2. Select the Virtual Hosts folder in the tree pane of the Admin Console.

3. In the Details pane, select the `default_host` virtual host.

4. Add all required new entries to the Host Aliases list of the `default_host` virtual host in the following format:
   
   `<Fully qualified Hostname of Remote WebServer>:`

   For our example installation this leads to two entries (see Figure 4-35 on page 218):
   
   - m10df55f.itso.ral.ibm.com:80
   - m10df55f.itso.ral.ibm.com:443
5. Regenerate the Web server Plugin. Right-click the node name and select the option **Regen Web server Plugin** as shown in Figure 4-28 on page 206.

6. Copy the `<WAS_HOME>/config/plugin-cfg.xml` file from the WebSphere Portal machine across to the `<WAS_HOME>/config` directory on the remote Web server machine.

    **Note:** If you chose for the WebSphere Application Server HTTP plug-in a different installation path on the remote server, you have to manually edit the plugin-cfg.xml file.

    In our example installation, we used the procedure as shown in Example 4-1 on page 219.
Example 4-1  Moving the plugin-cfg.xml file to the remote Web Server

# id
uid=0(root)  gid=0(system)
groups=2(bin),3(sys),7(security),8(cron),10(audit),11(lp)
# hostname
m10df4ff
# cd /usr/WebSphere/AppServer/config
# ftp m10df55f
Connected to m10df55f.itso.ral.ibm.com.
220 m10df55f FTP server (Version 4.1 Sat Feb 23 00:11:36 CST 2002) ready.
Name (m10df55f:root): root
331 Password required for root.
Password:
230 User root logged in.
ftp> cd /usr/WebSphere/AppServer/config
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> put plugin-cfg.xml
200 PORT command successful.
150 Opening data connection for plugin-cfg.xml.
226 Transfer complete.
10020 bytes sent in 0.002876 seconds (3402 Kbytes/s)
local: plugin-cfg.xml remote: plugin-cfg.xml
ftp> bye
221 Goodbye.
#

Enable the HTTP Server for SSL

Complete all the steps in 4.4.2, “Creating an SSL certificate” on page 196 and
4.4.3, “HTTP Server Setup” on page 199.

Change accessing host name information for WebSphere Portal

WebSphere Portal needs to be made aware which URL the users will use to
access the Portal pages.

- ConfigService.properties is located at
  <was_home>/lib/app/config/services/ConfigService.properties. The
  information of the property parameter host.name will be used to generate
  those URLs that are not generated by the server-relative.

  In our example, we change this property to the following value:

  host.name = m10df55f.itso.ral.ibm.com
Web.xml is located at <wp_home>/app/wps.ear/wps.war/Web-INF/Web.xml. The information in the <form-login-page> tag is the URL the user gets redirected to, if he wants to access a secured page without having a valid credential cookie. This usually happens when the user is not yet logged in.

In our example we changed this tag to the following value:

```
```

**Installation verification**

Before verifying that you have a correct installation, make sure you restart the IBM HTTP Server on the remote HTTP Server machine and restart the WebSphere Application Server node on the WebSphere Portal machine.

Use procedures as described above to do this.

Use a Web browser to verify the correct setup of SSL at your WebSphere Application Server. Request the portal public page with HTTP schema:

```
http://m10df55f.itso.ral.ibm.com/wps/portal
```

After switching to the Login page, you should be automatically redirected to get the HTTPS schema.

For a description of the verification test, see also 4.4.5, “WebSphere Portal Setup” on page 207.

### 4.6 Using External Security Manager

For information on using External Security Manager, please review the IBM Redbook, *Enterprise Business Portals with IBM Tivoli Access Manager Part II*, SG24-6885.
Site analysis

This chapter describes the support for WebSphere Site Analyzer available in IBM WebSphere Portal V4.1 to track logins, logouts, enrollments, errors, and portlet and page usage. A sample scenario using Site Analyzer V4 is included.
5.1 Introduction to Web site analysis

Setting up an Internet portal is an important step towards achieving one’s business goals. Web site analysis also helps in achieving these goals by reducing the cost of maintaining the site.

Thorough and frequent analysis of your Web site will provide very important information:
- Operational information, such as site performance, health, and usage
- Business information, such as customer demographics and content relevance

5.2 WebSphere Site Analyzer: An overview

Note: The version of WebSphere Site Analyzer that we refer to is V4.1. This version ships on disk #10 of the WebSphere Portal CDROM set.

The IBM WebSphere Site Analyzer is an enterprise-level Web analytical tool that transforms random Web data into valuable e-business intelligence. It captures, analyzes, and stores data, and generates reports on a Web site about the following:
- Usage
- Health
- Integrity
- Content

Site Analyzer collects information in two different ways:
- Content analyses are used to crawl a Web site starting at a particular URL. This collects information related to a Web site and its resources, such as resource size and structure, link information, and transfer rates.
- Usage analyses collect information from Web server logs. Information from a usage analysis reflects activity at the site, for example, who the users were, what pages they visited, and errors that occurred.

In the case of WebSphere Portal, the information is retrieved from the logs and the database, and a utility transforms this data into a Web server log format. These analyses can be run on a one-time basis or scheduled to run on a regular basis.

Data collected by the analyses is stored in the Site Analyzer database for use in reports. The reports produced by Site Analyzer come in many flavors. The report designer can choose among a variety of output formats. Reports can be static or
dynamic. Dynamic reports can be scheduled to be generated and published at a certain time, at repeated intervals, or whenever the associated analysis is run.

**Note:** We will create a sample report for usage analysis later in this chapter.

WebSphere Site Analyzer supports multi-channel data capture from a wide variety of sources:

- **Server logs**
  
  Server logs can be generated by:
  
  - IBM HTTP Server
  - WebSphere Application Server
  - WebSphere Personalization
  - WebSphere Portal
  - WebSphere Edge Server
  - WebSphere Commerce Suite

- **External files or databases**

- **Virtual real-time page information via Web Tracker**

Figure 5-1 on page 224 depicts the data import formats supported by WebSphere Site Analyzer.
Figure 5-1 Data import formats supported by Site Analyzer

However, please note that certain servers are capable of generating logs in only specific formats and Site Analyzer allows you to import only those formats. For example, when you choose to import a WebSphere Portal log file into the Site Analyzer database, you only have the option to import a log file in NCSA Combined format.

Web Tracker

Web Tracker is a data collection method in Site Analyzer that uses single-pixel technology to provide near real-time information about site usage. When Web Tracker is enabled, usage information is automatically sent directly from your user's browser to Site Analyzer for immediate processing.

The tool complements log file analysis in that it can provide:

- Faster access to usage data than that provided by log files
- Near real-time information that log file analysis cannot provide
- Tracking of very large sites where log file analysis is impractical
- Business data tracking
Enabling your Web site for Web Tracker Analysis would require you to include the Web Tracker JavaScript file in your Web pages. You would then need to switch Web Tracker to On in your Site Analyzer project.

The YourCoHotel application (http://<your.server>/SiteAnalyzer/Samples/YourCoHotel/index.jsp) that is installed as a part of the Site Analyzer installation (Site_Analyzer_Samples Enterprise Application) is a Web Tracker enable application. You can use this application as a reference for your implementation.

5.3 Reporting possibilities

Figure 5-2 shows the usage possibilities of reports generated with Site Analyzer.

<table>
<thead>
<tr>
<th>Traffic Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic volumes on visitors, repeat visitors,</td>
</tr>
<tr>
<td>registered customers</td>
</tr>
<tr>
<td>Traffic trends hour of day, day of week</td>
</tr>
<tr>
<td>Time spent by visitors on pages viewed</td>
</tr>
<tr>
<td>Effectiveness of referrers (advertisements and links on partner's web sites)</td>
</tr>
<tr>
<td>Visitor network geography (domains)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commerce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Products seen/selected</td>
</tr>
<tr>
<td>Shopping behavior (searching or browsing)</td>
</tr>
<tr>
<td>Shopping cart abandon rate</td>
</tr>
<tr>
<td>Campaign effectiveness</td>
</tr>
<tr>
<td>Business impact measure</td>
</tr>
<tr>
<td>Conversion rates (browse-to-buy, search-to-buy)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Path Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popular paths through the site</td>
</tr>
<tr>
<td>Content effectiveness</td>
</tr>
<tr>
<td>Effectiveness of specific site features</td>
</tr>
<tr>
<td>Customer needs (search analysis)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Personalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer profiling</td>
</tr>
<tr>
<td>Effectiveness of business rules &amp; campaigns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Site Operation Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server utilization measurements based on traffic data</td>
</tr>
<tr>
<td>Web site bugs (broken links, server errors)</td>
</tr>
<tr>
<td>Resource request-response time</td>
</tr>
</tbody>
</table>

Figure 5-2  Reporting possibilities with Site Analyzer
5.3.1 Portal reports

Site Analyzer provides the following report elements that are specific to WebSphere Portal:

- Portal Server Page Ranking - Displays a ranking of the Portal Server Pages viewed by visitors to your site.
- Portal Server Page Trend - Displays the Portal Server Pages viewed by your visitors over time.
- Portal Server Portlet Ranking - Displays a ranking of the Portal Server Portlets viewed by visitors to your site.
- Portal Server Portlet Trend - Displays the Portal Server Portlets viewed by your visitors over time.
- Portal Server Login Trend - Displays the Portal Server logins over time.
- Portal Server Login by User Ranking - Displays a ranking of the users who access your site using the Portal Server Login command.
- Portal Server Command Trend - Displays the Portal Server Commands used by your visitors over time.
- Portal Server Summary Trend - Displays summary statistics about Portal Server logs over time.
- Portal Server Page Edit Ranking - Displays a ranking of Portal Server Pages by the frequency with which they have been edited.
- Portal Server Page Edit by User Ranking - Displays a ranking of users by the frequency with which they have edited Portal Server Pages.

These elements can be used in conjunction with other report elements, such as those for the HTTP server, to create a Web site report.

5.3.2 Benefits

The following are the benefits of using Site Analyzer:

- A solution that provides a complete picture of the site
- Reports at application level, beyond HTTP logging
- Reporting promotes quick Web site and business reactions
- Tight integration with WebSphere and Portal family
- Flexible standard reports
- Customized reports using Report Elements as building blocks
- Real-time data feeds
- Open database schema for data warehousing and analysis
5.4 Planning

The following information has been provided for planning purposes.

5.4.1 Supported platforms

- Server platforms
  - AIX v4.3 or later
  - Solaris v2.6 or later
  - Linux (Red Hat, SuSE)
  - Windows 2000, NT

- Client browser platforms
  - Netscape 4.7+
  - IE 5.0+

- Languages
  - English, Spanish, French, German, Italian, Japanese, Korean, Simplified Chinese, Traditional Chinese, and Brazilian Portuguese

5.4.2 Prerequisites

Before you can install Site Analyzer V4.1, you must install the following software:

- DB2 UDB Version 7.2 with FixPack 5 or Oracle 8.1.7

  **Note:** If you want to connect to a remote DB2 database using the Net driver (COM.ibm.db2.jdbc.net.DB2Driver), the remote database must be at the exact same version level as your Site Analyzer server database.

- WebSphere Application Server Advanced Edition 4.0.2

5.4.3 Disk space considerations

Depending on the details that are being logged and the Web site traffic, the log files generated by different servers may span anywhere between a few kilobytes to a few hundred megabytes. Accordingly, you might need to back up the log files to external storage and remove them from the production environment to free up resources.
In WebSphere Portal, the logger can be configured to change to a new log file every few minutes/hours/days. The details will be discussed in 5.6.3, “Configuring logging for WebSphere Portal” on page 242.

5.4.4 Database considerations

Site Analyzer uses three databases for storing information:

- Administrative database (saadmin)
  The administrative database is used to store metadata and other information that Site Analyzer needs to operate.

- DNS/IP database (sadns)
  The DNS/IP database is used to store IP addresses and DNS information.

- Project database (saprojct)
  The project database is used to store data that is collected by Site Analyzer as it analyzes a site. It can include data from log file analysis, Web tracker analysis, or database analysis.

During installation, you can choose to use a single database for storing all the above information. Ideally, it is advisable to have three different databases. The project database stores the data captured by Site Analyzer and thus, demands the highest amount of resources.

As multiple log files are imported into the database, over a period of time the database can have a huge amount of information that may not be required. This data will have to be manually flushed out or backed up.

Also, if you plan to install Site Analyzer and Portal on the same server, it would be a good idea to create the Site Analyzer databases on a separate server.

5.4.5 Application Server considerations

Site Analyzer can be installed on the same WebSphere Application Server installation that hosts WebSphere Portal. However, in a production scenario, it is advisable to have Site Analyzer on a separate WebSphere Application Server.

If Site Analyzer and Portal are installed on the same WebSphere Application Server, then data imports and report generation should be scheduled during off-peak traffic hours.
5.4.6 Remote file system considerations

Site Analyzer can import log files from local as well as remote servers. It uses the File Transfer Protocol (FTP) to retrieve log files from remote servers. To enable data imports from remote servers, you would need to set up an FTP user account on all such servers. The user has to be granted permissions to read (get) files from the server log directory.

Similarly, reports can be published to local as well as remote file systems. To allow Site Analyzer to publish to a remote server, you would need to set up an FTP user account with write (put) permission to a directory on the server. It might be worthwhile to consider publishing reports to a directory from where an HTTP server can serve these files, for example, the htdocs directory for IBM HTTP Server or Apache server. Of course, the HTTP Server should only allow the Administrator or intranet users to view the server reports.

5.5 Installation using Portal Setup Manager

This section briefly covers Site Analyzer installation using Setup Manager. It is assumed that the prerequisites (see 5.4.2, “Prerequisites” on page 227) have already been installed.

**Important:** The Setup Manager does not create the database(s) required by Site Analyzer. So, if you are installing Site Analyzer as a part of the Portal installation, ensure that you have created the Site Analyzer databases before the virtual application server on which the Site Analyzer Enterprise Application is installed is started.

**Note:** For our sample scenario, we have chosen to create only the administrative database for Site Analyzer. The DNS/IP and project information will be stored in the same database.

We installed Site Analyzer on an existing WebSphere Application Server V4.02 (with DB2) installation on Linux. We had already applied the required fixpacks and e-fixes to the server and security was enabled. The server uses Domino V5.0.8 as its LDAP user registry. We have created a user, “saadmin”, with password, saadmin, in the LDAP directory. This user ID would be the Administrator ID for Site Analyzer.

Some steps of this installation might vary depending upon the platform on which Site Analyzer is being installed.
5.5.1 Creating the Site Analyzer administrative database

The Site Analyzer Project database must be configured to allow it to perform well even with a large amount of data. The administrative database is generally not as demanding in terms of the resources. However, since the administrative database in our sample scenario doubles as the Project database also, we will configure it per the configuration requirements of the latter.

1. Log in as the DB2 instance owner, for example “db2inst1” on UNIX and “db2admin” on Windows. Ensure that the instance is running by issuing the command `db2start` and then, start the command line utility (DB2).

2. Issue the following commands in sequence to create the database:
   
   ```
   db2 => UPDATE DBM CFG USING JAVA_HEAP_SZ 4096
   db2 => CREATE DATABASE saadmdb USING CODESET UTF-8 TERRITORY US COLLATE USING IDENTITY
   db2 => UPDATE DB CFG FOR saadmdb USING APPLHEAPSZ 8192 LOGPRIMARY 20 LOGSECOND 20 LOGFILSIZ 2000 DBHEAP 4096 STMTHEAP 4096 CATALOGCACHE_SZ 256 LOCKLIST 1024
   ```

   **Note:** The recommended DB2 configuration parameters for all three Site Analyzer databases can be found in the product InfoCenter in the *Install & Configure Site Analyzer* section.

3. Now we need to create an alias for the database we just created. To do this, issue the following commands:
   
   ```
   db2 => CATALOG TCPIP NODE sanode REMOTE m23vnx55 SERVER db2cdb2inst1
   db2 => CATALOG DB saadmdb AS saadmin AT NODE sanode
   ```

4. Close the command line utility by typing `quit`.

5. You should now restart the DB2 instance. Please note that you should also close all applications that are currently accessing the database so that you can safely restart the database server. Restart the database server by issuing the command `db2stop` and then the command `db2start`.

6. You can verify the database configuration by connecting to the database. To do this, issue the command:
   
   ```
   db2 connect to saadmin user db2inst1 using ibmdb2
   ```

   Once the connection has been established successfully, disconnect by issuing the command:
   
   ```
   db2 disconnect current
   ```
5.5.2 Installing Site Analyzer

Following are the steps that you would allow you to setup the Site Analyzer server:

1. Insert the Setup Manager CD (disk #1) and start the installer by issuing the command `/mnt/cdrom/install.sh`.

2. You are shown the Welcome window with a link to the prerequisites. Click Next.

3. Accept the license agreement and then click Next.

4. Specify the install key and click Next.

5. Select Standard Installation as the install type and click Next.

6. You are asked to provide a response file from a previous install. Click Next.

7. From the list of components to install, select WebSphere Site Analyzer. Setup Manager automatically select the prerequisites. However, since we have the current versions of the prerequisites installed, Setup Manager will not try to install those products. Click Next.

8. Setup Manager will check for the products that have already been installed. Click Next.

9. We have already enabled security for WAS. So, select Yes when asked Is WAS security enabled? and then click Next.

10. You are asked to provide the user ID and password for WebSphere Application Server security. Specify the ID and password, for example, "wpsbind/wpsbind" and click Next.

11. Setup Manager will require you to provide the path to the directory where WebSphere Application Server has been installed. Also, you will need to specify the directory where you want the Site Analyzer files to be copied. We retain the default values (see Figure 5-3 on page 232) and then click Next.
12. Site Analyzer can either be installed on an already existing (virtual) application server or a new one. However, it is recommended that you install it on a new, separate application server. Select **Create New Server** and then click **Next**.
13. You are now asked when you want to configure Site Analyzer security. Select **Now** and click **Next**.

14. Then, you are required to specify the security ID and password used for WebSphere Application Server. Also you can specify the security ID (saadmin) and password (saadmin) for the Site Analyzer application. See Figure 5-5 on page 234. Retain the defaults and click **Next**.
15. Specify the access settings for the Site Analyzer Administrative database (saadmin). Click **Next**.
16. For the DNS/IP database in a production environment, you can choose the **Use an existing database** option. If you choose this option, the next window allows you to specify the settings for the DNS/IP database. For this scenario, choose the **Use the Site Analyzer administrative database** option and click **Next**.
17. Similarly, for the production database in a production environment, you can choose **Use an existing database**. For this scenario, choose **Use the Site Analyzer administrative database** and click **Next**.

---

**Figure 5-7  Settings for DNS/IP database**
18. Setup Manager displays a summary of the Site Analyzer installation. Click **Next** to begin installation.
19. Setup Manager will ask you to insert the WebSphere Site Analyzer CD (disk #10). Change the discs and click **OK**.

20. Once Setup Manager has finished installing Site Analyzer, it will display a pop-up window (see Figure 5-10 on page 239). Click **OK** to close this window and then click **Finish** to close Setup Manager.
Before you can start the Site Analyzer application server, you need to set up the administrative users for the Site Analyzer Enterprise Application that has been installed. To do this, follow these instructions:

1. Open the WebSphere Administration Console.

2. From the list of Enterprise Applications on the left side of the console, select Site_Analyzer_Application.

3. On the right side of the console, click User/Role Mappings.

4. Highlight the SiteAnalyzer Admin role and then click Select.

5. Choose the Select users/groups option. Specify the wildcard * as the search pattern and then click Search. Add the required users from the list of Available Users/Groups and then click OK. See Figure 5-11 on page 240.
6. Click **Apply** to save the changes.

7. Right-click WebSphere Application Server node, for example **m23vnx55**, on the left side of the console, and click **Regen Webserver Plugin**. It is advisable to restart the WebSphere Application Server and the HTTP Server before you start the Site Analyzer application server.

### 5.6 Using Site Analyzer

Site Analyzer provides a Web-based GUI for importing data and generating reports. This interface enables secure, remote administration for Site Analyzer.

Generating reports in Site Analyzer is a two-step process:

1. Import data, usually from server log files, into the database

   The administrator first needs to enable logging for the required servers. The servers may have to be reconfigured to create log files in formats supported by the Site Analyzer.
2. Create and publish reports

The reports can be either be scheduled (nightly/weekly) or be generated on demand. The Administrator can be notified by e-mail of errors and/or warnings while importing data or generating reports.

The next few subsections discuss the details involved in using Site Analyzer to generate reports for WebSphere Portal.

5.6.1 Configuring NCSA Combined logging for IBM HTTP Server

Site Analyzer supports the W3C Extended, NCSA Separate, NCSA Combined and NCSA Common log file formats for HTTP server logs. However, in our sample scenario we used the NCSA Combined format, since this is the format used by the WebSphere Portal logging module. To configure IBM HTTP Server for NCSA Combined logging, follow these instructions:

1. Stop the IBM HTTP Server.
2. Open the file <HTTPServer_ROOT>/conf/httpd.conf for editing.
3. By default, IBM HTTP Server uses the NCSA Common format. Comment out this line in the file so that it looks like this:

```
#CustomLog logs/access_log common
```

**Note:** By default, the access log file is named access_log on UNIX platforms and access.log on Windows.

4. Find the line for enabling combined logging and uncomment it.

```
CustomLog logs/access_log combined
```

You might want to change the name of the file from access_log to something else. If you do not do this, then you should delete/rename the access_log file that already exists.

5. Save the changes and then start IBM HTTP Server.

**Important:** The log file is empty until the HTTP server is accessed.

5.6.2 Configuring logging for WebSphere Personalization

Since we are not using the Personalization log files in our scenario, this subsection is provided as an aside. Depending upon the type of applications running on WebSphere Portal, you might not need to analyze the Personalization log file even though Personalization Server has been installed.
There are three logging options for WebSphere Personalization:

- File logging - to a flat file format
- Database logging - to a relational database
- Web Tracker - directly to WebSphere Site Analyzer in real time

This is also called “HTTP Logging”, because it uses real-time HTTP requests, not the HTTP log file.

Web Tracker is preferred for the situations where real-time data is desired. The other two formats require you to import the data into Site Analyzer before reports can be created.

However, here, we will enable file logging because it is similar to the HTTP and Portal logging formats:

1. Open the WebSphere Personalization Resource Console, for example, http://your.hostname.com/wps/PersAdmin/adminframe.jsp.
2. Click the Log Settings tab.
3. Select the Use File Logging option and specify a log file, for example, /opt/PortalServer/log/Pzn.log.
4. Select Enable Rule Logging.
5. Click Save. You would need to restart WebSphere Application Server to start logging. The log file would be empty unless there is some activity on the Personalization engine, for example, rule execution.

### 5.6.3 Configuring logging for WebSphere Portal

Perform the following instructions to configure logging:

1. Open the file WAS_root/lib/app/config/jlog.properties for editing.

   **Important:** We have chosen to enable logging modules for only certain Portal events. However, all supported Site Analyzer logging modules can be enabled by simply removing the comment from the following line and then proceeding to Step 9 on page 244:

   ```
   baseGroup.SiteAnalyzerLogger.isLogging=true
   ```

   More information on the individual modules can be found in the Site Analysis section of the WebSphere Portal InfoCenter.

2. Find the group of properties that starts with SiteAnalyzerLogService and un-comment the following line so that it looks like this:
Chapter 5. Site analysis

baseGroup.SiteAnalyzerLogTraceLogger.isLogging=true

3. Find the baseGroup.SiteAnalyzerFileHandler sub-group so that it looks something like this

```
baseGroup.SiteAnalyzerFileHandler.filename=log/sa.log
baseGroup.SiteAnalyzerFileHandler.dateFormat=yyyy.MM.dd-HH.mm.ss
#baseGroup.SiteAnalyzerFileHandler.minutesPerLogFile=1
#baseGroup.SiteAnalyzerFileHandler.hoursPerLogFile=1
baseGroup.SiteAnalyzerFileHandler.daysPerLogFile=10
```

The parameter baseGroup.SiteAnalyzerFileHandler.dateFormat controls the name of the log files that are backed up at specified intervals. The value you specify is appended to the base log file name to form the backup file name.

To control the interval at which the log file is backed up, set the dateFormat parameter for only one of the following options:

- If you want to log in intervals of minutes, uncomment
  baseGroup.SiteAnalyzerFileHandler.minutesPerLogFile and set the value to an integer in the range 1 to 60.
- If you want to log in intervals of hours, uncomment
  baseGroup.SiteAnalyzerFileHandler.hoursPerLogFile and set the value to an integer in the range 1 to 24.
- If you want to log in intervals of days, uncomment
  baseGroup.SiteAnalyzerFileHandler.daysPerLogFile and set the value to an integer that indicates the number of days.

If you enable more than one date format interval, the smallest interval will be used.

In case of a high traffic Web site, the file should be backed up every few hours to limit the file size. For our sample scenario, we have set a rather long interval of 10 days between backups.

4. Find the section for logon/logoff events and remove the comment from the following line:

baseGroup.SiteAnalyzerSessionLogger.isLogging=true

5. Find the new users section and remove the comment from the following line:

baseGroup.SiteAnalyzerUserManagementLogger.isLogging=true

6. Find the section for logging rendering of pages and remove the comment from the following line:

baseGroup.SiteAnalyzerPageLogger.isLogging=true
7. Find the section for logging rendering of portlets and remove the comment from the following line:

   baseGroup.SiteAnalyzerPortletLogger.isLogging=true

8. Find the section for logging errors when rendering portlets/pages and remove the comment from the following line:

   baseGroup.SiteAnalyzerErrorLogger.isLogging=true

9. Save the file and restart the Portal application server. The file would be created when the first configured event is logged. The file, sa.log, can be located in the <PORTAL_ROOT>/log directory.

5.6.4 Creating a Site Analyzer project

This sub-section explains the creation of a sample project:

1. Open the Site Analyzer GUI from the location
   http://<your.server.domain>/SiteAnalyzer/Admin/loginIn.jsp. The browser asks for a user name and password. This will be the administrative user that you have created during Site Analyzer installation. Click OK.

2. If you do not have any projects created, Site Analyzer will start the Project wizard. You can choose to exit the wizard and then add a project by clicking the Add Project button. Using the Add Project option allows us to specify all configuration parameters for the project at creation time. If a project is created using the wizard, then we might have to edit the project later on. We chose to use the wizard. Click Next.

3. Provide a name for your project and click Next.

4. Provide the host names for the Web site. You might want to include the IP addresses and network IDs and the site host machine(s). Click Next. You will see a window similar to Figure 5-12 on page 245.
5. Specify the access parameters for the Site Analyzer projects database. Note, that we had chosen to use the administration database (saadmin) to store the data for projects. In a production environment, you would have a separate project database (saproj). Click **Next**. You will see a window similar to Figure 5-13 on page 246.
6. Click **Finish** to create the project.

7. Now, we need to change some configuration parameters for our project, which we could not specify while using the wizard. Select the project from the list of projects on the left side of the browser window and then click the **Edit Project** button. You will see a window similar to Figure 5-14 on page 247.
8. Open the **URL Parameters** tab to choose the URL parameters that will be used to collect data for Web applications such as JSPs, servlets, or CGIs you have implemented on your Web site. Select **All URLs**.
9. Open the **Referral Parameters** tab to collect parameter data on referrals. Select **All URLs**.

10. Open the **Cookie Keys** tab and select **All cookie keys**.
11. Open the **User ID** tab (see Figure 5-16). The User ID setting is used to tell Site Analyzer how to recognize the User ID field in the log records. Select option 6. **Use Custom key=value Pair Field** and specify the Key Name as UserId.

12. Click **Save** to save the changes.

13. Now we need to specify the e-mail server settings to be used by Site Analyzer to send notifications and reports. On the main admin window, click **Global Settings**.

14. In the Global Settings pop-up window, click **Email Server**. Specify the SMTP Email Server and the Return Mail Address. Click **Save**.

### 5.6.5 Importing log files into Site Analyzer

This section covers the task of importing data from the HTTP Server and Portal log files into a Site Analyzer project. You can import log files into Site Analyzer either using the Log File wizard or by clicking the **Add Log File** button. We will import the HTTP server log by using the wizard and the Portal log using the latter method, just to get a feel of it.
1. Open the Site Analyzer GUI from the location
   http://<your.server.domain>/SiteAnalyzer/Admin/loginIn.jsp. Specify
   the user ID and password of an administrative user and click OK.

2. On the left side of the browser window, select the project that you created by
   clicking it. On the right side, click Data Imports tab if it hasn’t been already
   displayed.

3. Click the Log File Wizard button to specify the import settings for the HTTP
   server log file. On the Welcome window, click Next.

4. On the Log File Information window, provide the following values (see
   Figure 5-17):
   – Name: HTTP log (can be anything)
   – Type: HTTP Server
   – Syntax: NCSA Combined
   – Log File Name: access_log (or whatever you specified in 5.6.1,
     “Configuring NCSA Combined logging for IBM HTTP Server” on page 241)

   ![Log File Wizard - Log file information window](image)

   Figure 5-17  Log File Wizard - Log file information window

   Click Next.
5. On the Log File Location window, specify the log file location as Remote (see Figure 5-18). If your HTTP server log file is local to the Site Analyzer server, then you would select Local. In that case, the next window would be different and would require you to specify the local path to the file.

![Log File Wizard - Log file location window](image)

Figure 5-18 Log File Wizard - Log file location window

6. Click Next. You will see a window similar to Figure 5-19 on page 252.
7. On the Location Information window, provide the FTP settings for transferring the log file. For example,
   - Host Name: m10df55f.itso.rai.ibm.com
   - Directory: /opt/IBMHTTPServer/logs
   - User ID: saadmin
   - Password: saadmin

   Click Next. You will see a window similar to Figure 5-20 on page 253.
8. On the Schedule Log File window, retain the default values and click **Next**. You will see a window similar to Figure 5-21 on page 254.

![Log File Wizard - Schedule Log File window](image.png)

**Figure 5-20**  Log File Wizard - Schedule Log File window
9. On the Email Notification window, you can choose (not required) to be notified of errors/warnings or success. Select **Errors** and **Warnings** and specify an e-mail ID on which you would like to receive notifications. Click **Next**.

**Note:** In order for the e-mail notification feature to work properly, you need to have specified a valid SMTP server and e-mail ID in the Email Server section of the Global Settings for Site Analyzer.

10. On the confirmation window, click **Finish** to schedule the file import and return to the main administration window. Clicking **Refresh** periodically would display the current status of the import.
11. Now, import the Portal log file by clicking the **Add Log File** button.

12. You will be shown the Log File Information tab, which includes the Log File Location section.

   Provide the following values for the first section:
   
   - **Name**: Portal log
   - **Type**: Portal Server
   - **Syntax**: NCSA Combined (this is the only option for Portal logs)
   - **Log File Name**: sa*.log

   For the Log File Location part, select **Local** or **Remote** as appropriate. If choosing Remote, provide the FTP settings for accessing the log file.
13. Open the **Schedule Log File** tab. Set the Run After Save? field to **Yes** and retain the default values for all other fields.
14. Open the **Email Notification** tab and set up Site Analyzer e-mail notification similar to the one we set up for the HTTP log file import. Click **Save** to schedule the import.

15. The Log Files list in the Site Analyzer Data Imports section should show both the HTTP log and Portal log. Clicking the status of any of the log files would display the status monitor. See Figure 5-25 on page 258 for a sample.
5.6.6 Creating a sample Portal report

In this section, we quickly step through the creation of a sample report for our portal:

1. Open the Site Analyzer GUI from the location
   http://<your.server.domain>/SiteAnalyzer/Admin/loginIn.jsp. Specify
   the user ID and password of an administrative user and click OK.

2. On the left side of the browser window, select the project into which you have
   imported the HTTP server and Portal log files. On the right side, click the
   Reports tab.

3. Click the Add Report button.

4. The Report Information tab will be shown (see Figure 5-26 on page 259):
   – Provide a name for the report, for example, Portal Report.
   – Select the database for your project from the list of available databases.
   – Check the dates in the Report Range.
5. Open the **Report Elements** tab. The list of report elements will be empty. Follow the steps below to add some elements.
   - Click the **Add** button (refer to Figure 5-27 on page 260).
   - From the Report Element Group drop-down list, select **Portal Server Usage**. Note that you can also select other element groups, for example HTTP Server. The data for HTTP Server report elements has already been imported in the database.
   - Select the desired elements from the list of Report Elements. Note that when you click an element, you will be shown a short description of the element at the bottom of the window.
   - Click **Save**.
Figure 5-27   Add Report Elements - Portal Server Usage elements

The Report Elements tab displays the elements that we just added. You can change the order (vertical) in which the elements appear in your report by clicking the Up and Down buttons on the left side of the list. See Figure 5-28 on page 261.
6. Open the **Publishing Options** tab. You can choose to publish the report to one to three destinations (see Figure 5-29 on page 262).

   - **File System** - publish to the local file system.
     
     We chose this option and published the report to the htdocs directory from where our HTTP server could serve the report.
   
   - **FTP** - publish to a remote file system using FTP.
   
   - **Email** - generate report and send the HTML output as an attachment to an e-mail account.

![Add Report - Report Elements tab](image-url)
7. Open the **Schedule Report** tab. Select the **Run after Save?** option.

8. The Email Notification tab is similar to the Email Notification tab that we discussed in Step 9 on page 254.

9. Click **Save** to schedule report generation.

10. After a few minutes, click the **Refresh** button on the Site Analyzer Admin Console window. Once the Status field for the Portal Report changes to Complete, locate the file in the publish destinations that you chose in Step 6 on page 261 and open it in a Web browser. See Figure 5-30 on page 263 for the output of our sample report.
Figure 5-30  Our sample report in a Web browser
## Abbreviations and acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2B</td>
<td>Business-to-Business</td>
</tr>
<tr>
<td>B2C</td>
<td>Business-to-Customer</td>
</tr>
<tr>
<td>B2E</td>
<td>Business-to-Employee</td>
</tr>
<tr>
<td>CRM</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>CVS</td>
<td>Credential Vault system</td>
</tr>
<tr>
<td>CVS</td>
<td>Concurrent Versions System</td>
</tr>
<tr>
<td>DIIOI</td>
<td>Domino Internet Inter-ORB Protocol</td>
</tr>
<tr>
<td>DMT</td>
<td>Directory Management Tool</td>
</tr>
<tr>
<td>DN</td>
<td>Distinguished Name</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>EIP</td>
<td>Enterprise Information Portal</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise JavaBeans</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GNOME</td>
<td>GNU Network Object Model Environment</td>
</tr>
<tr>
<td>GNU</td>
<td>UNIX-like operating system</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines Corporation</td>
</tr>
<tr>
<td>IHS</td>
<td>IBM HTTP Server</td>
</tr>
<tr>
<td>IIOP</td>
<td>Internet Inter-ORB Protocol</td>
</tr>
<tr>
<td>INSO</td>
<td>IntraNet Solution</td>
</tr>
<tr>
<td>IPSec</td>
<td>Internet Protocol Security</td>
</tr>
<tr>
<td>ITSO</td>
<td>International Technical Support Organization</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform, Enterprise Edition</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>JDK</td>
<td>Java Development Kit</td>
</tr>
<tr>
<td>JDNI</td>
<td>Java Naming and Directory Interface</td>
</tr>
<tr>
<td>JRE</td>
<td>Java Runtime Environment</td>
</tr>
<tr>
<td>JSP</td>
<td>JavaServer Pages</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>KDE</td>
<td>K Desktop Environment</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>LTPA</td>
<td>Lightweight Third Party Authentication</td>
</tr>
<tr>
<td>LUM</td>
<td>License Use Management</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>RDN</td>
<td>Relative Distinguish Name</td>
</tr>
<tr>
<td>RPM</td>
<td>Red Hat Package Manager</td>
</tr>
<tr>
<td>SASL</td>
<td>Simple Authentication and Security Layer</td>
</tr>
<tr>
<td>SCM</td>
<td>Supply Chain Management</td>
</tr>
<tr>
<td>SMIT</td>
<td>System Management Interface Tool</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign-On</td>
</tr>
<tr>
<td>TAI</td>
<td>Trust Association Interceptor</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>WCM</td>
<td>WebSphere Content Manager</td>
</tr>
<tr>
<td>WCP</td>
<td>Web Content Publisher</td>
</tr>
<tr>
<td>WML</td>
<td>Wireless Markup Language</td>
</tr>
<tr>
<td>WMS</td>
<td>WebSphere Member Services</td>
</tr>
<tr>
<td>WPS</td>
<td>WebSphere Portal</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>Extensible Stylesheet Language Transformations</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2003. All rights reserved.
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 269. Note that some of the documents referenced here may be available in softcopy only.

- *Deploying QuickPlace*, SG24-6535
- *Customizing QuickPlace*, SG24-6000
- *Lotus Discovery Server 2.0: Deployment, Planning, and Integration*, SG24-6575
- *Inside the Lotus Discovery Server*, SG24-6252
- *WebSphere Portal Collaborative Components*, REDP0319
- *IBM WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883
- *IBM WebSphere Portal V4.1 Handbook Volume 2*, SG24-6920
- *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176
- *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885

Other publications

These publications are also relevant as further information sources:

Online resources

These Web sites and URLs are also relevant as further information sources:

- Lotus Domino Workflow
  http://www.lotus.com/products/domworkflow.nsf
- InfoCenter - Lotus Workflow
- WebSphere Personalization
- WebSphere Portal
- Lotus Developer Domain
  http://www-10.lotus.com/ldd/
- Domino 5.0.8 Release Notes
  http://doc.notes.net/uafiles.nsf/docs/rn508/$File/readme.pdf
- Sametime 2.5 Release Notes
- Sametime Installation Notes
  http://doc.notes.net/uafiles.nsf/docs/ST25/$File/stinstall.pdfs
- QuickPlace Installation Notes
  http://doc.notes.net/uafiles.nsf/docs/QP208/$File/QPAdminBP.pdf
- Juru - Full-text search library
  http://www.haifa.il.ibm.com/km/ir/juru/
- IBM Corporation
- WebSphere Application Server - InfoCenter
  http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/0606080004aa.html
How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks
Index

Numerics
3A product 166

A
Access Control
   permission types 173
      copy 174
      delegate 174
      edit 173
      manage 174
      view 173
   resource 176
      group 177
      groups 176
   rules 173
      object 173
      permissiontype 173
      subject 173
Access Control functionality 169
Access Control List 165
Access Control List (ACL) portlet 169
Access Control List portlet 171
AddModule statements 200
admin role list 182
administration 165
administration component 168
administrative database 230
Administraive group DN 82
Administrator DN 82
AIX environment 194
assigning permissions 181
associated tables 135
authentication 165
authentication component 166
authentication proxy 167
authoring server 109
authoring template 5
authoring templates 3
authorization 165
authorization component 168

B
back-end systems 182

C
caching proxy 216
certifier organization 22
certifier password 22
classpath 138
ClearModuleList statement 200
client 167
Client-Web App SSO 167
Collaboration portlets 149
Collaborative Places 9
common object request broker architecture (CORBA) 166
Configuring logging for WebSphere Personalization 241
Configuring logging for WebSphere Portal 242
Configuring NCSA Combined logging for IBM HTTP Server 241
ccontent analyses 222
ccontent contributor 4, 105
ccontent publisher 105
ccontent templates 141
ccontext root 109–110
CORBA 166
ccrawler.properties 154, 162
ccreate portlet 181
Creating a sample Portal report 258
Creating a Site Analyzer project 244
Credential Vault (CV) 183
Credential Vault concept 167
Credential Vault segments 183
Credential Vault Service 193
Credential Vault slots 183, 189
Credential Vault system 165, 182–183, 185
CSEnvironment.properties 152
CVS 8

data types 4
data validation 4
database table 99
DB2 administrators 28
DB2 Universal Database 10, 83
default workflows
  simple change process 105
  simpler change process 105
  simplest change process 105
delegate 174
detail view 6
DIIOP 23
Discovery Server 148
DNS/IP database 228
document search 154
Document Search portlet 154–155
Documentum 164
domain expert 105
Domain name 22
Domino 9–10, 15, 148, 167
Domino administrator 32
Domino administrator password 22
Domino administrators 56
Domino clients 26
Domino components 15
Domino Directory Services 23, 94
Domino LDAP 32, 71

e
edit mode 173
editions 8
e-fixes 215
e-mail notification 254
Enterprise Application 93
Enterprise Information Portal 164
extended search 153
external access control 179
external authorization manager 175
external version control 8

F
Federated search 163
field lengths 4
File Transfer Protocol (FTP) 229
file-level locking 8
Filenet 164
firewall 195, 211
formats 4
FTP 8

g
generation templates 4, 6, 144
detail view 6
  summary view 6
Global Security Toolkit 18, 79
Group DN prefix 82
Group DN suffix 82
Group Member 82
Group Object Class 82
groups 171

H
hostname 22
HTML 4, 110, 164
HTML page 3
https schema 201

I
IBM HTTP Server 10, 79, 199
IBM Key Management 196
index 156
INSO technology 164
Installation using Portal Setup Manager 229
Internet address 70
Internet domain 70
Internet mail account 36
Internet portal 222
IntraNet Solutions (INSO) 164
Introduction to Web Site Analysis 222
IPSec 195

J
J2EE servlet 142
Java AgentRunner 49
Java APIs 134
JNDI name 132
JSP 4, 110, 120

L
LDAP 23, 166
LDAP directory structure 168
LDAP server 9
LDAP user registry 229
lightweight directory access protocol (LDAP) 166
lightweight third party authentication (LTPA) 12, 166
log file wizard 249
Lotus Architect 15
Lotus Collaboration portlets
Lotus QuickPlace 149
My iNotes 149
My Notes Calendar 149
My Notes Mail 149
My Notes To Do 149
Notes Discussion 149
Notes Mail 149
Notes View 149
Sametime Chat 149
Teem room 149
Lotus Collaborative Components 147
types
   Java Classes and Methods 148
   JavaScript tag libraries 148
Lotus Collaborative Places 147
Lotus Developer Domain 152
Lotus Domino Extended Search 163
Lotus Notes client configuration 33
Lotus Notes databases 185
Lotus QuickPlace 151
Lotus Sametime 151
Lotus Workflow 15, 87
default workflows 105
managing 107
Lotus Workflow 3.0 Architect 32
LTPA 166
LTPA cookie 166
LTPA File 25
LTPA password 25
LWD Organization R3.0 95
LWF Application R3.0 43, 51, 95
LWF Design Repository R3.0 43, 95
LWF Organization R3.0 43, 107
LWF Organization R3.0.01-1 50
LWF Process Definition R3.0 43, 95

M
manage portal 181
Manage Search Index portlet 154, 156–158
members 101
meta-data 112

N
name and address book 103
NCSA Combined format 224
NCSA Combined log file format 241
NCSA Common log file format 241
NCSA Separate 241
Netegrity Siteminder 169
Notes Organization Directory 43

O
Oracle 95

P
pages 177
permission types 173–174
PersAdmin 99
Personalization
   rule editor 137
personalization engine 138
personalize e-mail 99
places 177
Planning 227
POP3 mail box 185
Portal Administration 186
Portal administrators 167
portal configuration interface 180
Portal reports 226
portal search 153
Portal security 165
Portal-Back End SSO 167
portlet 167–168
portlet API 182
portlet applications 177
portlet catalog 164
portlet private credential slot 190
portlets 79, 85, 148, 178
PortletService interface 182
preview templates 6
primary table 135
Process Definition database 43
production database 236
production server 7
productivity portlets 86
project database 228, 230
project files 111
project root directory 110
proxy server 38
proxy settings 143
publish server 143–144
publishing environment 7–8
Web Server 23
Web SSO configuration document 89
Web Tracker 224
Web Tracker JavaScript file 225
WebApplication 113
WebSphere Application Server 10, 79, 167, 182, 195
WebSphere LTPA keys 92
WebSphere Personalization 16, 79, 139, 241
WebSphere Personalization Resource Console 139, 242
WebSphere Personalization Resource Hierarchy 139
WebSphere Portal 9, 56, 78, 167, 175, 184
  installation 78
  log files 249
WebSphere Portal Extend 147
WebSphere Portal log file 224
WebSphere Portal permission 176
WebSphere Portal Security whitepaper 267
WebSphere Portal Setup Manager 229
WebSphere Site Analyzer 221, 223, 240
  benefits 226
  planning 227
  report elements
    Portal server command trend 226
    Portal server login by user ranking 226
    Portal server login trend 226
    Portal server page edit 226
    Portal server page edit ranking 226
    Portal server page ranking 226
    Portal server page trend 226
    Portal server portlet ranking 226
    Portal server portlet trend 226
    Portal server summary 226
    Portal server summary trend 226
  security 233
WebSphere Site Analyzer - An Overview 222
WebSphere Studio Application Developer 8, 138
WML 4
workflow participants 101
workflow participants group 103
workflow process 46

X
XML 112, 164
xmlaccess tool 180
IBM WebSphere Portal V4.1 Handbook Volume 3
IBM WebSphere Portal V4.1 Handbook
Volume 3

Understand the IBM WebSphere Portal architecture

The IBM WebSphere Portal V4.1 Handbook is available in three volumes of Redbooks. This is Volume 3.

These IBM Redbooks position the IBM WebSphere Portal for Multiplatforms as a solution that provides a single point of interaction with dynamic information, applications, processes and people to help build successful business-to-employee (B2E), business-to-business (B2B), and business-to-consumer (B2C) portals.

WebSphere Portal consists of three packaged offerings:
- Portal Enable
- Portal Extend
- Portal Experience

In the three volumes of the IBM WebSphere Portal V4.1 Handbook, we cover WebSphere Portal Enable and Extend.

The IBM WebSphere Portal V4.1 Handbook will help you to understand the WebSphere Portal architecture, teaches you how to install and configure WebSphere Portal, discusses how to administer portal pages using WebSphere Portal and the development of WebSphere Portal portlets, and covers how to use specific WebSphere Portal applications.

In this redbook, we discuss the WebSphere Portal applications and their uses.

Step-by-step installation instructions for IBM WebSphere Portal

Implement new and enhanced capabilities of IBM WebSphere Portal

For more information:
ibm.com/redbooks