



**DEPLOYMENT GUIDE**  
**FOR THE**  
**PLUMTREE ENTERPRISE WEB**  
**SUITE 5**

Version 6

May 2005

Information in this document is subject to change without notice.

Companies, names, and data used in examples herein are fictitious unless otherwise noted.

© 2001-2005 Plumtree Software. All rights reserved.

Plumtree Software may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property rights except as expressly provided in any written license agreement from Plumtree Software.

Full licensing information is available in the files:

**PortalAttributions.txt; OEWCAttributions.txt; ContentServerAttributions.txt; CollaborationServerAttributions.txt; and StudioServerAttributions.txt**, on the Plumtree Corporate Portal 5.0 and higher; Plumtree Optional Enterprise Web Components 5.0.2 and higher; Plumtree Content Server 5.0 and higher; Plumtree Collaboration Server 3.0 and higher; and Plumtree Studio Server 2.0 and higher installation CD images.

The Plumtree software ("Software") described and covered in this document is commercial computer software developed exclusively at private expense, and in all respects is proprietary data belonging solely to Plumtree or its suppliers. (a) Department of Defense End Users. (i) If the Software is acquired by or on behalf of agencies or units of the Department of Defense (DoD), then, pursuant to DoD FAR Supplement Section 227.7202 and its successors (48 C.F.R. 227.7202) the Government's right to use, reproduce or disclose the Software acquired under this Agreement is subject to the restrictions of the Software License Agreement between the Licensee and Plumtree. (b) Civilian Agency End Users. (i) If the Software is acquired by or on behalf of civilian agencies of the U.S. Government, then, pursuant to FAR Section 12.212 and its successors (48 C.F.R. 12.212), the Government's right to use, reproduce or disclose the Software acquired under this Agreement is subject to the restrictions of the Software License Agreement between the Licensee and Plumtree.

SiteMinder® is a registered trademark of Netegrity, Inc.

Oblix, NetPoint, Oblix NetPoint, and the Oblix logo are registered trademarks of Oblix, Inc. NetPoint COREid System: User Manager, Group Manager, Organization Manager, IdentityXML, Certificate Processing Server (VeriSign®), COREid Server, and WebPass; NetPoint Access System: Access Manager, Access Server, WebGate, and AccessGate; COREid, FEDERATEDid Layer, Oblix IDLink, Associate Portal Services, NetPoint System Console, NetPoint Ready Realm, NetPoint Federation Services, NetPoint Mainframe Security Connector, NetPoint SAML Services, and their logos are trademarks of Oblix, Inc.

Java™, Sun™ and any other Sun Microsystems product or technology are trademarks or registered trademarks of Sun Microsystems, Inc.

BEA WebLogic® and any other BEA products are trademarks or registered trademarks of BEA Systems, Inc.

IBM WebSphere® and any other IBM products are trademarks or registered trademarks of IBM Corporation.

Oracle® is a registered trademark of Oracle Corporation.

Microsoft® Internet Information Server (IIS), Microsoft® SQL Server, Microsoft® Exchange, Excel®, Microsoft® Pocket PC and other Microsoft products are trademarks or registered trademarks of Microsoft Corporation.

ColdFusion®, JRun™ and any other Macromedia® products are trademarks or registered trademarks of Macromedia, Inc.

Documentum® is a registered trademark of Documentum, Inc. All rights reserved.

DOCS Open® is a registered trademark of Hummingbird Ltd. All rights reserved.

Lotus Notes® and other Lotus products are trademarks or registered trademarks of Lotus Development Corporation.

SAP® software and other SAP products are trademarks or registered trademarks of SAP AG.

Siebel™ and any Siebel products are trademarks of Siebel Systems, Inc., and might be registered in certain jurisdictions.

Infomentum and ActiveFile are trademarks of Infomentum Ltd.

Jaws® for Windows® is a registered trademark of Freedom Scientific™ BLV Group

AvantGo® and any other AvantGo products are trademarks or registered trademarks of AvantGo, Inc.

BlackBerry™ and RIM™ are trademarks of Research In Motion Limited.

Openwave™ and Phone.com™ are trademarks of Openwave Systems Inc. or its subsidiaries in the United States and/or other countries.

Palm™ is a trademark of Palm, Inc.

Sprint PCS is a service mark of Sprint Communications Company L.P.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Regular Expression support is provided by Regexp regular expression Java Library Copyright (c) 2000 the Apache Software Foundation.

Regular expression support is provided by the PCRE library package, Copyright (c) 1997-2000 University of Cambridge.

This product includes software distributed under the terms of the Sun Binary Code License Agreement. See, for example, [http://java.sun.com/j2se/1.3/jre/j2re-1\\_3\\_1\\_01.license.html](http://java.sun.com/j2se/1.3/jre/j2re-1_3_1_01.license.html).

Wsd4j.jar is provided by IBM under the terms of the Common Public License (<http://oss.software.ibm.com/developerworks/opensource/CPLv1.0.htm>).

Copyright 2003, IBM Corporation. All rights reserved.

ICU 2.1 - International Components for Unicode Copyright © 1995-2001: ICU 1.81 and Later Copyright and Permission Notice. Copyright (c) 1995-2001 International Business Machines Corporation and others. All rights reserved.

Saxon XSLT Processor developed by Michael Kay, is made available under the terms of the Mozilla Public License, version 1.1. <http://www.mozilla.org/MPL/MPL-1.1.html>

HTTP Client Library developed by Ronald Tschalar is made available under the terms of the GNU Lesser General Public License (LGPL): <http://www.gnu.org/licenses/lgpl.html>.

Log4Net, Copyright 2001-2003 Neoworks, Ltd. All rights reserved, is made available under the terms of the Apache Public License.

JavaService, Copyright (c) 2000, Alexandria Software Consulting. <http://www.alexandriasc.com/software/JavaService/index.html>

GLUE is used under license from The Mind Electric. Copyright © The Mind Electric. All rights reserved. <http://www.themindelectric.com/glue/index.html>

The source code, object code, and documentation in the com.oreilly.servlet package is licensed by Hunter Digital Ventures, LLC.

LDAP client © Netscape, distributed under the terms of the Netscape/iPlanet ONE SDK.

This product uses SmartClient technology licensed from Isomorphic. Copyright © 2003, Isomorphic Software, Inc. All rights reserved.

The Hoard Multiprocessor Memory Allocator (<http://www.hoard.org>). Copyright © 1998 - 2003, The University of Texas at Austin.

Threading is provided by the POSIX Threads Library for Windows Copyright © 1991, 1999 Free Software Foundation, Inc.

The Standard Template Library is provided by STLport Copyright © 1999, 2000 Boris Fomitchev. Copyright 1994 Hewlett-Packard Company. Copyright 1996,97 Silicon Graphics Computer Systems, Inc. Copyright 1997 Moscow Center for SPARC Technology.

Dom4j, Copyright © MetaStuff, Ltd. All Rights Reserved. Thanks to the Dom4j project, <http://dom4j.org/>.

This product includes software developed by the JDOM Project (<http://www.jdom.org/>).

Copyright (C) 2000-2003 Jason Hunter & Brett McLaughlin. All rights reserved.

This product uses the Opta 2000 JDBC driver under license from I-net Software, GMBH. Copyright © 2000-2003 I-Net Software, GMBH. All rights reserved.

Oracle JDBC driver distributed by permission of Oracle Corporation. Copyright © 2003 Oracle Corporation. All rights reserved.

This product uses JUG, by Tatu Saloranta, distributed here under the terms of the LGPL license. JUG is available here: <http://www.doomdark.org/doomdark/proj/jug/index.html>

## Copyright

Servlet support classes are provided by com.oreilly.servlet Copyright (c) 2001 by Jason Hunter <jhunter@servlets.com>. All rights Reserved.

File conversion functionality provided by Outside In® Viewer Technology© 1992-2001 Stellent Chicago, Inc. All rights reserved.

XML parsing is provided by XP - an XML Parser in Java Version 0.5 Copyright(c) 1997, 1998 James Clark.

GNU sed version 3.02 Copyright © 1998 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE, to the extent permitted by law.

Navigation menus powered by HierMenus (<http://www.webreference.com/dhtml/hiermenus>) by DHTML Lab(<http://www.dhtmlab.com/>), © Peter Belesis 2001

Outside In® Viewer Technology© 1992-2001 Stellent Chicago, Inc. All rights reserved.

The online help for this product includes portions copyright © eHelp Corporation. All rights reserved.

Other product and company name mentioned herein may be the trademarks of their respective owners.

If you have comments on this document, send e-mail to: [documentation@plumtree.com](mailto:documentation@plumtree.com).

# Table of Contents

I	Introduction .....	I-1
	Typographical Conventions Used in This Book .....	I-2
	Icons Used in This Book .....	I-3
	Plumtree Documentation and Resources .....	I-4
2	Business Foundation .....	2-1
	What is the Enterprise Web? .....	2-2
	Elements of the Enterprise Web .....	2-2
	The Architecture of the Enterprise Web .....	2-4
	Traditional Infrastructure .....	2-4
	Integration Products .....	2-5
	Foundation Services .....	2-6
	Portal Platform .....	2-7
	Plumtree Analytics Server .....	2-8
	Composite Applications .....	2-9
	Enterprise Web Business Drivers .....	2-10
	Enterprise Web Benefits .....	2-10
	Deployment Strategies .....	2-11
	Deployment Best Practices .....	2-12
	Take Advantage of Plumtree Resources .....	2-13
	Supporting Different Audiences .....	2-15
	Should You Have Multiple Portals? .....	2-15
	User Views .....	2-16
	Defining Subportals .....	2-17
	Defining Communities .....	2-18
	Branding Subportals and Communities .....	2-27
	Controlling User Views with Access Privileges .....	2-28
	Securing and Managing the Enterprise Web .....	2-29
	Access Privileges .....	2-30
	Mandatory Administrators .....	2-31
	Special Cases .....	2-31
	Activity Rights .....	2-31
	Defining Roles .....	2-32
	Common Activity Rights Roles .....	2-33
	Defining an Administrative Object Hierarchy .....	2-35
	Providing Access to Divergent Sources of Content .....	2-37
	Types of Data .....	2-37

	Location of Data .....	2-38
	File Repositories .....	2-38
	Discovering Content .....	2-40
	Defining a Knowledge Directory Taxonomy .....	2-41
	Managing Content .....	2-44
	Defining Search .....	2-45
	Crawler Web Services Versus Search Web Services .....	2-46
	Understanding What Users Are Searching For .....	2-49
	Influencing What Users Find .....	2-50
	Related Materials .....	2-51
	Providing Interactive and Informational Applications to Users Through Portlets .....	2-52
	Why Build Portlets? .....	2-52
	Which Portlets to Build First .....	2-53
	Building Good Portlets .....	2-53
	Creating a Rich Enterprise Web Experience .....	2-55
	What Are Enterprise Web Applications? .....	2-55
	Portal User Interfaces .....	2-57
	Sourcing from Common Sources .....	2-59
	Sourcing from Custom Sources .....	2-62
	Example 1: Access and Personalization .....	2-62
	Example 2: Searching for Data and Documents .....	2-63
	Example 3: Customer Branded Support Site .....	2-63
3	Technology Foundation .....	3-1
	Configuration Options .....	3-1
	Components Involved in an Enterprise Web Suite Deployment .....	3-5
	Web Applications .....	3-5
	Static Web Components .....	3-10
	Services .....	3-11
	Database Services .....	3-14
	Remote Servers .....	3-14
	Consolidating Components .....	3-17
	One-Machine Configuration .....	3-19
	Omitting Components .....	3-20
	Two-Machine Configuration .....	3-21
	Notes and Guidelines .....	3-21
	Risk and Risk Mitigation .....	3-22
	Three-Machine Configuration .....	3-23
	Variation 1: Smaller Number of Users, Larger Amount of Content .....	3-23

Variation 2: Larger Number of Users, Smaller Amount of Content .....	3-25
Four-Machine Configuration .....	3-26
Variation 1: Medium Users, Medium Content .....	3-26
Variation 2: Higher Availability, More Users, Less Content .....	3-28
Variation 3: Large Amounts of Knowledge Directory Content .....	3-30
Five-Machine (or More) Configuration .....	3-32
Hardware Sizing and Scaling .....	3-34
Portal Server Sizing and Scaling .....	3-34
Estimating Portal Server Traffic .....	3-35
Benchmark Data for Portal Server Capacity in a Windows Deployment .....	3-37
Benchmark Data for Portal Server Capacity in Unix Deployments .....	3-40
Portal Server Capacity and Availability .....	3-41
Database Sizing and Scaling .....	3-43
Search Server Sizing and Scaling .....	3-44
Search Server Scaling and Performance .....	3-46
Search Server Configuration File Sizing .....	3-47
General Assumptions About the Search Server Environment .....	3-50
Administrative Portal Server Sizing and Scaling .....	3-50
Automation Server Sizing and Scaling .....	3-51
Image Server Scaling .....	3-53
Remote Server Sizing and Scaling .....	3-54
Custom Services Sizing and Scaling .....	3-55
Portlet Server .....	3-55
Authentication Server .....	3-55
Profile Web Service Server .....	3-56
Search Web Service Server .....	3-56
Crawler Web Service Server .....	3-57
Using the Plumtree Remote Client (PRC) .....	3-57
Collaboration Server Sizing and Scaling .....	3-58
Content Server Sizing and Scaling .....	3-58
Studio Server Sizing and Scaling .....	3-60
Analytics Server Sizing and Scaling .....	3-60
Scaling Using Federated Portals .....	3-62
High Availability Deployment .....	3-62
Portal Server Load Balancing .....	3-62
Database Server Load Balancing .....	3-63
Search Server Load Balancing .....	3-63
Automation Server Load Balancing .....	3-65
Remote Server Load Balancing .....	3-65
Collaboration Server Load Balancing .....	3-65

Content Server Load Balancing.....	3-67
Studio Server Load Balancing.....	3-67
Analytics Server Load Balancing .....	3-67
Document Repository Load Balancing .....	3-67
External Service Load Balancing .....	3-68
Parallel Portal Engine (PPE).....	3-68
Security Strategies.....	3-83
Configuring SSL for Your Enterprise Web Deployment .....	3-83
Portal Security Modes.....	3-84
Roadmap to Changing Security Modes and Setting Up SSL .....	3-85
Importing CA Certificates into the cacerts Keystore (for Java Portals) ..	3-87
Importing CA Certificates into MMC (for .NET Portals) .....	3-88
Setting Up Content Server to Use a Secure Image Server or Portal Server .	3-89
Setting Up Workflow Server to Use a Secure Image Server or Portal Server	3-90
Setting Up Collaboration Server to Use a Secure Image Server .....	3-91
Setting Up Studio Server to Use a Secure Image Server or Portal Server .	3-91
Troubleshooting.....	3-92
Search Server Security .....	3-94
Collaboration Server Security .....	3-94
Content Server Security .....	3-95
Studio Server Security .....	3-95
Analytics Server Security .....	3-95
The DMZ (Demilitarized Zone) .....	3-96
Firewalls and Security .....	3-97
Implementing the Plumtree Corporate Portal in a DMZ.....	3-98
Perimeter Networks (DMZ) .....	3-98
Enterprise Web Components and Their Communication Protocols .....	3-98
Web Services and Internal Network Security .....	3-103
Extranet Strategies .....	3-104
Authentication Source Provider Configuration .....	3-104
Risk Mitigation Scenarios.....	3-105
Encryption .....	3-107
Secure Sockets Layer (SSL) .....	3-107
Encryption of Persistent Data.....	3-109
PKI and Digital Certificates .....	3-109
About Public Key Infrastructure (PKI) .....	3-109
Delegation and Portals .....	3-111
Using PKI in Your Portal.....	3-113



Summary .....	3-114
Single Sign-On Options .....	3-115
What SSO Means in the Enterprise Web .....	3-115
Logging in to the Portal .....	3-115
Logging In to the Portal with Auto-Authentication .....	3-116
Brokering Credentials to the Remote Tier .....	3-118
Summary .....	3-119
Architecture for Development, Staging, and Production Environments .....	3-121
Portal Server Migration .....	3-121
Search Server Migration .....	3-121
Remote Server Migration .....	3-122
Collaboration Server Migration .....	3-123
Content Server Migration .....	3-123
Logs .....	3-124
Studio Server Migration .....	3-125
Analytics Server Migration .....	3-125
Internationalization .....	3-126
Multilingual User Interface .....	3-126
Unicode Support .....	3-126
Localizing Names and Descriptions of Objects Stored in the Database .....	3-127
Adding a User Interface Language to the Portal .....	3-129
Adding Language Style Sheets .....	3-129
Adding an Online Help Language .....	3-132
Adding Javascript Language Files .....	3-132
Language Support in the Knowledge Directory .....	3-133
<b>4 Business Implementation .....</b>	<b>4-1</b>
Crawler Best Practices .....	4-1
Plumtree Crawler Web Service for Microsoft Exchange .....	4-1
Plumtree Crawler Web Service for Lotus Notes .....	4-2
Plumtree Crawler Web Service for NT File Systems .....	4-3
Plumtree Crawler Web Service for Documentum .....	4-3
Building Crawlers to Other Enterprise Systems .....	4-4

Portlet Best Practices .....	4-6
Subportal Best Practices .....	4-7
Community Best Practices .....	4-8
Incorporating Collaboration Server .....	4-9
Sales Process Automation .....	4-9
Customer Support .....	4-10
Finance Application .....	4-10
E-learning Application .....	4-11
How Not to Use Collaboration Server? .....	4-11
Incorporating Content Server .....	4-12
The Content Server Portlet Templates .....	4-13
Using Content Server to Create Workflow Applications .....	4-14
Content Server Limitations .....	4-16
Similarities and Differences Between Content Server and Studio Server .....	4-17
Related Materials .....	4-18
Incorporating Studio Server .....	4-19
The Studio Server Frameworks/Templates .....	4-19
Sharing Databases Across Studio Server Portlets .....	4-20
Using Studio Server to Create Workflow Applications .....	4-21
Accessing Data in the Studio Server Database Tables .....	4-22
Studio Server Limitations .....	4-23
Incorporating Analytics Server and Analytics Server Portlets .....	4-23
Incorporating Branding (Customizing the User Interface) .....	4-25
Subportals .....	4-25
Navigation .....	4-25
Style Sheets and Portlets .....	4-26
Branding .....	4-26
Pluggable Event Interfaces (PEIs) .....	4-27
Custom Activity Spaces .....	4-27
Putting It All Together .....	4-27
Common Requests and Solutions .....	4-29
Rolling Out the Enterprise Web .....	4-34
5 Production Maintenance .....	5-1
Production Maintenance Checklist .....	5-1
Troubleshooting Tools .....	5-2
Performance Monitor Counters .....	5-2

Performance Monitoring Uses .....	5-3
Available Performance Monitor Counters .....	5-4
Running Performance Monitor Counters .....	5-9
Performance Monitoring Suggestions .....	5-10
Monitoring Databases and Java Application Servers .....	5-13
Plumtree Analytics Server .....	5-13
PTSpy .....	5-17
Using PTSpy .....	5-18
Runtime Settings .....	5-20
PTSpy Output .....	5-21
View Source .....	5-22
When to Use View Source .....	5-22
How to Use View Source .....	5-22
What Is Available in View Source .....	5-23
ODBC Testing .....	5-24
When to Use ODBC Testing .....	5-24
ODBC Testing: SQL Server .....	5-24
ODBC Testing: Oracle .....	5-26
Testing and Troubleshooting Enterprise Web Content .....	5-28
General .....	5-28
Crawlers .....	5-30
Portal Search .....	5-31
Portal Search/Snapshot Queries Portlet .....	5-31
Search Server Maintenance .....	5-34
Logs .....	5-34
System Events .....	5-34
Search Server Logs .....	5-34
Log Rotation .....	5-34
Status Monitoring .....	5-35
Performance Monitoring .....	5-35
Backup and Restore .....	5-36
Repair .....	5-36
Index Synchronization with Portal .....	5-36
Index Corruption - Self Repair .....	5-36
Index Corruption - Manual repair .....	5-37
Rebuild Search Index .....	5-37



# Introduction

The Plumtree Deployment Guide aggregates resources to assist customers, partners, and consultants with deploying Enterprise Web technology. Use this guide to create a solid Enterprise Web strategy, so you can use Plumtree's most powerful features to create a stable, useful Enterprise Web solution that can grow with your company without having to be reorganized or recreated every time the company changes.

The resources contained within the Deployment Guide include guidelines and processes that complement each phase of the Plumtree Deployment Methodology:

- Business Foundation, in which you align Enterprise Web technology to your corporate objectives. [Chapter 2 “Business Foundation”](#) is written for business sponsors (the people driving the Enterprise Web project) and the select company experts who understand your business needs.
- Technology Foundation, in which you align your network and hardware to your Enterprise Web, making sure that it performs, scales and provides adequate security for your business. [Chapter 3 “Technology Foundation”](#) is written for IT personnel tasked with installing and configuring the Plumtree Enterprise Web Suite.
- Business Implementation, in which you put your Enterprise Web plan into action. [Chapter 4 “Business Implementation”](#) is written for business sponsors (or the people tasked with encouraging portal adoption) and Enterprise Web administrators.
- Production Maintenance, in which you maintain the Enterprise Web and monitor its performance. [Chapter 5 “Production Maintenance”](#) is written for Enterprise Web administrators and/or IT personnel tasked with maintaining the Enterprise Web.

The Deployment Guide is updated as additional content is assembled based on implementation best practices and lessons learned through real-world implementation.



**Important:** The Deployment Guide is a complementary tool to other Plumtree Documentation, and is not a replacement for installation, developer, or user guides. In addition, the Deployment Guide is not a substitute for formal Education Services courses or Plumtree Consulting Services offerings. For more information, refer to [“Plumtree Documentation and Resources” on page I-4.](#)

# Typographical Conventions Used in This Book

This book uses the following typographical conventions.

Table 1-1: Typographical Conventions

Convention	Typeface	Example
<ul style="list-style-type: none"><li>• File names</li><li>• Folder names</li><li>• Screen elements</li></ul>	<b>bold</b>	<ul style="list-style-type: none"><li>• Upload <b>Procedures.doc</b> to the portal.</li><li>• Open the <b>General</b> folder.</li><li>• To save your changes, click <b>Apply Changes</b>.</li></ul>
<ul style="list-style-type: none"><li>• Text you enter</li></ul>	<code>computer</code>	<ul style="list-style-type: none"><li>• <code>Type Marketing</code> as the name of your community.</li></ul>
Variables you enter	<i>italic</i> <code>computer</code>	Enter the base URL for the Remote Server. For example, <code>http://my_computer/</code> .
<ul style="list-style-type: none"><li>• New terms</li><li>• Emphasis</li><li>• Plumtree object example names</li></ul>	<i>italic</i>	<ul style="list-style-type: none"><li>• <i>Portlets</i> are Web tools, embedded in your portal.</li><li>• The URI <i>must</i> be a unique number.</li><li>• The example Knowledge Directory displayed in Figure 5 shows the <i>Human Resources</i> folder.</li></ul>

## Icons Used in This Book

This book uses the following margin icons:



**Note:** The Note icon is used to denote tips, best practices, or additional information related to the content in a paragraph.



**Security:** The Security icon is used to denote important information related to security for your portal.



**Important:** The Important icon is used to denote important information (including warnings) related to the content in a paragraph.

# Plumtree Documentation and Resources

This section describes the documentation and resources provided by Plumtree.

Table 1-2: Plumtree Documentation and Resources (Sheet 1 of 3)


Resource	Description
Installation and Upgrade Guides	These guides are written for users who will install or upgrade Enterprise Web components. They are available with the installer packages and in the Plumtree Support Center (described later).
Release Notes	These files are written for IT personnel and/or Enterprise Web administrators. They include information about new features and known issues in the release. They are available with the installer packages and in the Plumtree Support Center (described later).
Administrator and User Guides	These guides are written for different levels of Enterprise Web users. They describe how to use the Enterprise Web user interface, including conceptual information and best practices. They are available with the installer packages and in the Plumtree Support Center (described later).
Developer Guides, Quickstarts, API Documentation, and Sample Code	These documents are written for developers. They describe how to customize the Enterprise Web user interface and features. They are available with the product installer packages and/or in the Plumtree Support Center (described later).
Online Help	<p>Online help is written for all levels of Enterprise Web users. It describes the Enterprise Web user interface, including detailed instructions for completing tasks in the Enterprise Web.</p> <p>To access online help, click  <b>Help</b> in the upper-right corner of the user interface.</p>



Table 1-2: Plumtree Documentation and Resources (Sheet 2 of 3)

Resource	Description
Plumtree Support Center	<p>The Plumtree Support Center is a comprehensive repository for technical information on Plumtree products. From the Support Center, you can access product documentation, search knowledge base articles, read the latest news and information, participate in a support community, get training, and find tools to meet most of your Plumtree-related needs. The Support Center encompasses the following communities:</p> <p><b>Technical Support Center</b></p> <p>Submit and track support incidents and feature requests, search the knowledge base, and download service packs and hotfixes.</p> <p><b>Deployment Center</b></p> <p>Find the tools you need to roll-out, drive, and maintain a successful Enterprise Web deployment. Collaborate with peers on strategic business and technical objectives, learn application best practices, download launch examples, and calculate your return on investment (ROI).</p> <p><b>Product Center</b></p> <p>Download products, read Release Notes, and access recent product documentation.</p> <p><b>Developer Center</b></p> <p>Download developer tools and documentation, get help with your development project, and interact with other developers via discussion forums.</p> <p><b>Education Center</b></p> <p>Find information about available training courses, purchase training credits, and register for upcoming classes.</p> <p>If you do not see the Support Center when you log in to <a href="http://portal.plumtree.com">http://portal.plumtree.com</a>, contact <a href="mailto:support@plumtree.com">support@plumtree.com</a> for the appropriate access privileges.</p>

Table 1-2: Plumtree Documentation and Resources (Sheet 3 of 3)

Resource	Description										
Technical Support	<p>If you cannot resolve an issue using the above resources, Plumtree Technical Support is happy to assist. Our staff is available 24 hours a day, 7 days a week to handle all your technical support needs.</p> <p>E-mail: <a href="mailto:support@plumtree.com">support@plumtree.com</a></p> <p>Phone Numbers:</p> <table><tr><td>U.S. and Canada</td><td>+1 415.263.1696 or +1 866.262.PLUM (7586)</td></tr><tr><td>Asia Pacific</td><td>+61 2.9931.7822</td></tr><tr><td>Europe and U.K.</td><td>+44 (0)1628 589124</td></tr><tr><td>France</td><td>+33 1.46.91.86.79</td></tr><tr><td>Singapore</td><td>+65 6832.7747</td></tr></table>	U.S. and Canada	+1 415.263.1696 or +1 866.262.PLUM (7586)	Asia Pacific	+61 2.9931.7822	Europe and U.K.	+44 (0)1628 589124	France	+33 1.46.91.86.79	Singapore	+65 6832.7747
U.S. and Canada	+1 415.263.1696 or +1 866.262.PLUM (7586)										
Asia Pacific	+61 2.9931.7822										
Europe and U.K.	+44 (0)1628 589124										
France	+33 1.46.91.86.79										
Singapore	+65 6832.7747										

# 2

## Business Foundation

This chapter is written for business sponsors (the people driving the Enterprise Web project) and the select company experts who understand your business needs. It is designed to help you align Enterprise Web technology to your corporate objectives. It includes the following information:

- General overview of the Enterprise Web, [“What is the Enterprise Web?” on page 2-2](#)
- Suggested strategies for deploying the Enterprise Web, [“Deployment Strategies” on page 2-11](#)
- Common business problems and their Enterprise Web solutions:
  - [“Supporting Different Audiences” on page 2-15](#)
  - [“Securing and Managing the Enterprise Web” on page 2-29](#)
  - [“Providing Access to Divergent Sources of Content” on page 2-37](#)
  - [“Providing Interactive and Informational Applications to Users Through Portlets” on page 2-52](#)
- Using Enterprise Web solutions in conjunction with each other to provide powerful, integrated functionality to your users, [“Creating a Rich Enterprise Web Experience” on page 2-55](#)

## What is the Enterprise Web?



**Note:** This is a general overview of the Enterprise Web. You should familiarize yourself with the administrator and user guides and the online help of the available components to fully understand the features available to you.

The Enterprise Web is an open environment for managing and delivering Web applications. The Enterprise Web combines services from different vendors in a technology layer that spans rival platforms and business systems, creating a foundation for building applications at lower cost. This foundation consists of the services most commonly used by Web applications, including business integration, collaboration, content management, identity management, and search, which work together via web services. The portal is the delivery framework for applications created from this foundation.

The result is an environment that spans the entire enterprise, open to all platforms, and available to all audiences. Providing a common foundation for Web applications built on any platform lowers infrastructure and development costs; integrating resources from different systems into Web applications increases the return on those systems; and creating a common user experience for audiences across the enterprise to work together drives enterprise productivity and increases profits.

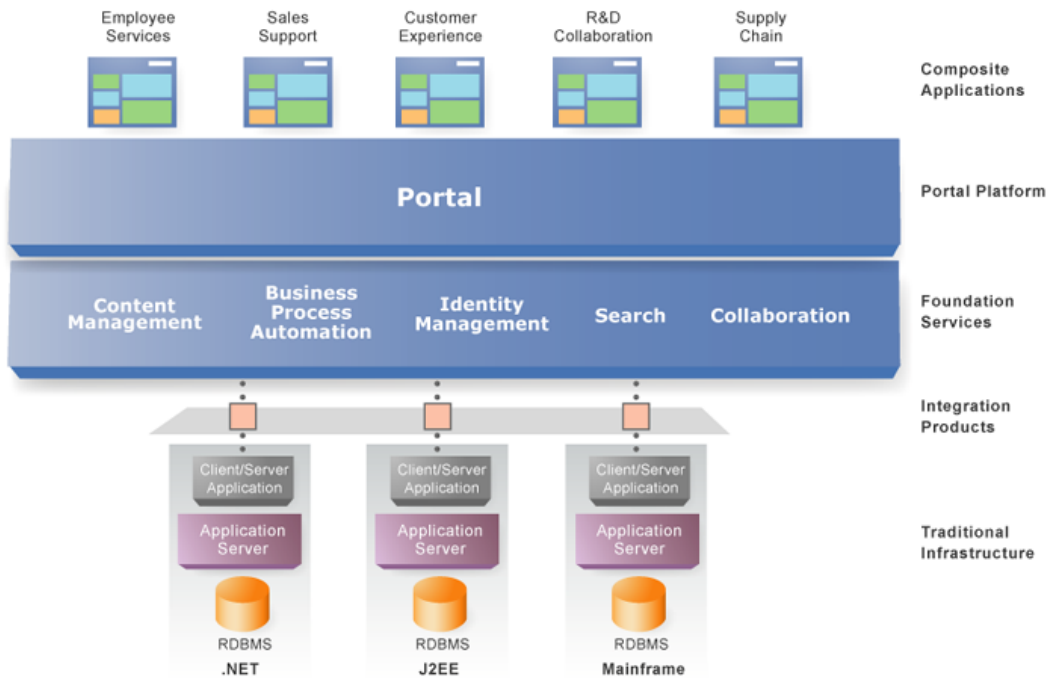
## Elements of the Enterprise Web

The Enterprise Web consists of four elements:

- **Foundation Services:** the core services commonly used to build Web applications, including business integration, collaboration, content management, identity management, and search. These services vary by deployment, and might include others not discussed here.
- **Integration Products:** components that integrate content, application services, and security from existing systems, such as Groupware, ERP, and CRM systems.

- **Portal Platform:** the personalization, administrative, and knowledge management framework for bringing together a broad range of electronic resources in different Web experiences.
- **Composite Applications:** Web applications hosted within the portal that combine services from different systems, which the portal orchestrates in custom business processes.

The following graphic shows how all the elements work together to form the Enterprise Web.



## The Architecture of the Enterprise Web

The Enterprise Web architecture is based on three principles:

- **Internet-Based:** Whereas client-server systems generally depend on local area network connections between a database, an application server, and an application, the Enterprise Web relies on the Internet's HyperText Transfer Protocol (HTTP), which can span networks and organizational barriers.
- **Cross-Platform:** Whereas client-server applications typically host every component of the application on a single application server, the Enterprise Web uses web services to combine components from different application servers, ensuring the Enterprise Web remains open to heterogeneous infrastructure.
- **Extensible:** Whereas client-server applications are based on their own business logic, which can only be changed at great effort and expense, the Enterprise Web is designed to be flexible and extensible, integrating other systems to maximize return-on-investment.

Each of the elements of the Enterprise Web is discussed later, but we begin with the traditional infrastructure on which the Enterprise Web is based.

### Traditional Infrastructure

Traditional infrastructure consists primarily of databases, application servers and applications. The value of the Enterprise Web lies in increasing the return on this infrastructure, not replacing it. Databases store their own data, but do not integrate data in other repositories. The majority of applications feature Web clients, but each only to present their own information and services. Application servers offer rich infrastructure for running programs developed in a single language, but are not designed to orchestrate components hosted in other environments.

The Enterprise Web differs from the traditional vertical stacks of databases, operating systems, and application servers by operating at right angles to these systems. The hallmark of the Enterprise Web is that it is open and horizontal:

- **Cross-Platform:** The technology of the Enterprise Web is web services, which can combine information and applications hosted on different application servers and operating systems.
- **Best-of-Breed:** The credo of the Enterprise Web is integration, building on, rather than replacing, existing systems.
- **Integrated:** The value of the Enterprise Web is greater than the sum of its parts, because previously disparate technologies work together to create a holistic user and administrative experience.

The Enterprise Web uses databases to store information, application servers to run software programs, and traditional applications to capture information and create business processes. But a monolithic approach that attempts to store all data in one relational database, run all programs on one application server, or route all business processes through one application is a legacy of the client-server era, which ignores the breadth and diversity of electronic resources relevant to the enterprise. To address this breadth, the Enterprise Web combines information from multiple databases, business processes from multiple applications, and web services from multiple application servers.

## Integration Products

Integration web services incorporate traditional systems such as SAP R/3 or Siebel eBusiness Applications into the broader confines of the Enterprise Web, bringing a greater return on the investment in those systems. Because the systems underlying the Enterprise Web run on different platforms, with varying reliability and performance, web services-based integration is the most practical approach, combining the functionality of traditional object-oriented protocols with the flexibility and fault-tolerance of Internet protocols. When integration services feature a graphical user interface that can be embedded in a portal, they are known as *portlets*. Other types of web services integrate content and security into the Enterprise Web from other systems.

## Foundation Services

As the technologies most commonly required to build a wide variety of Web applications, foundation services are designed to span traditional technical and organizational boundaries—for managing knowledge, supporting collaboration, and rationalizing security across a business. The foundation services deployed most consistently across the Enterprise Web are:

- **Business Process Automation:** Rather than creating business processes for every Web application in every business unit, a business integration engine acts as a foundation service for the entire Enterprise Web, routing transactions across many systems. As a result, the Enterprise Web not only surfaces information and services from many systems, but also creates business processes spanning those systems, driving productivity across the enterprise.
- **Content Management:** Organizations typically publish an enormous volume of content to different Web sites. Content management provides templates for capturing and publishing information in an organized way across the entire enterprise, with workflow and version-control for controlling what is published to the Enterprise Web, and a central repository for managing the information. Rather than creating a separate repository for every Web application, the Enterprise Web relies on a common content management system, enforcing consistent publishing policies and simplifying management of the data.
- **Collaboration:** The majority of Web applications are collaborative, with some facility for tracking projects, sharing documents, assigning tasks, exchanging ideas, or sending messages. By offering collaboration as a foundation service, the Enterprise Web can recognize dependencies between projects, and combine tasks and documents from different projects into a universal inbox for each user, allowing employees and customers to work together across traditional organizational and network boundaries.
- **Identity Management and Security:** Virtually every Web application authenticates users and develops its own identity profile of their roles and privileges. Deploying an identity management system as a foundation service for all the applications developed within the Enterprise Web allows users to navigate between Web applications without repeatedly logging in, and creates a common identity profile of user roles and privileges on which all the applications can draw. Identity management typically consists of a Lightweight Directory Access Protocol (LDAP) user directory, tools for administering that



directory, and a single sign-on system for issuing tokens to different systems. With an identity management system, security can be more consistently and easily enforced across the Enterprise Web, and applications are less expensive to develop.

- **Search and Categorization:** Search is deployed with almost every Web application. Deploying search as a foundation service within the Enterprise Web ensures that the content created by every Web application, the documents submitted to every collaborative project, the information routed through every business process can be easily found. Search can also index all the information stored in traditional systems in a knowledge management application such as the Knowledge Directory, with a set of folders that users can browse.

Some portal vendors, including Plumtree Software, offer search and collaboration as native portal services. But no vendor is the exclusive provider of these services. The open architecture of the Enterprise Web encompasses technologies of many vendors. This architecture is not only open to different technologies, but also extensible, so as to incorporate new types of technology into its foundation. While this section has described the foundation services most commonly used to create Web applications, you can extend this foundation to include services such as alerting, business process automation, or e-commerce.

## Portal Platform

Portal software provides a framework for creating and managing Enterprise Web applications, based on both foundation services and resources integrated from other systems. While many portals today allow administrators to create pages displaying elements from different applications alongside one another, a new generation of portals is emerging that can host not just simple pages, but sophisticated Web applications built from the components of the Enterprise Web. This entails support for:

- **Composite Applications:** The portal hosts different Web applications that have separate administrative domains and roles, a more flexible user interface, and installation templates.
- **Foundation Services:** The portal orchestrates foundation services as part of the application-building process, allowing portal administrators to quickly assemble new applications based on those services.

- **Business Process Integration:** The portal routes data between different foundation services and traditional systems, allowing business processes to span different technologies.
- **Usage Tracking:** The portal monitors the user's interests and activities across different applications, so that every resource integrated into the Enterprise Web can be delivered and customized to users based on that data. The portal contains the basic performance monitor counters described in [“Performance Monitor Counters” on page 5-2](#). For information on the advanced usage tracking portal add-on, the Plumtree Analytics Server, see [“Plumtree Analytics Server,”](#) next.

## Plumtree Analytics Server

Plumtree Analytics Server is an advanced usage tracking and analytics tool designed exclusively for the Plumtree portal. This portal add-on enables you to assess portal return-on-investment (ROI) and define future opportunities with usage trends in mind. Plumtree Analytics Server delivers the following features out-of-the-box:

- **Usage Tracking Metrics:** Analytics Server plugs directly into the portal engine to retrieve metrics for common portal functions, including community, portlet, and document hits, as well as search queries, logins, and more.
- **Behavior tracking:** Analytics Server tracks usage patterns, such as community visits, as well as the duration of portal use and community response times.
- **User Profile Correlation:** Analytics Server correlates metric information with user profile information in order for usage tracking reports to be viewed and filtered by profile data, such as country, company, and department.

Plumtree Analytics Server is designed to plug-n-play with the Plumtree portal suite. Information is gathered in real time using low overhead and the reliable Plumtree Message Bus (PMB) transport protocol. The reports can be retrieved via the portal in real time using either a secure or non-secure connection. Additionally, the Analytics Server star schema database approach ensures high report performance, as well as the ability to store millions of portal events.

## Composite Applications

Composite applications are Web applications built from the Enterprise Web's foundation services and elements integrated from traditional systems, which nonetheless work together as if hosted by a single system. An example of a composite application could be an employee services application that includes benefits and pension information from external providers, corporate communications announcements hosted by a content management system, policy and procedure documents, collaborative forums, and self-service capabilities hosted by PeopleSoft. A sales support site or a customer service center require a similar degree of integration, offering employees, partners and customers a simple way to get the most out of a wide range of systems deployed for their benefit.

The choice you face when attempting to deliver composite applications is whether to integrate other services into a Web-enabled version of traditional applications such as PeopleSoft and Siebel, or whether to draw elements from traditional applications into the Enterprise Web environment. As the number of composite applications increases, the rationale becomes stronger for an investment in Enterprise Web infrastructure as a common platform for all composite applications, rather than investments in extending each client-server application.

## Enterprise Web Business Drivers

Three trends are driving the formation of the Enterprise Web:

- **Strategic approaches to technology:** Rather than approving tactical investments in individual systems, companies increasingly favor a strategic approach to information technology, and want to understand how a wide range of Web technologies can work together across the enterprise.
- **Investments in Web infrastructure, web services:** Companies seek to invest in the infrastructure necessary to support their Web strategy, without casting aside what they already have. The client-server model of separate application servers and separate user experiences for each Web application has created too many cost centers in a business. New web services technologies can integrate existing technologies, without forcing you to re-build your entire business on a single platform.
- **Complete solution requirement:** Companies prefer the openness of a best-of-breed architecture, but seek a single solution for their Web strategy. The Enterprise Web offers both.

## Enterprise Web Benefits

The Enterprise Web addresses the most basic challenges faced by today's CIO:

- **Lower development costs:** How to integrate Web technologies to create a common foundation for Web applications, lowering development costs
- **Return on investment:** How to overcome client-server incompatibilities between Java, Windows, and legacy systems, maximizing the return on investments in existing systems
- **Enterprise productivity:** How to create, on the Web, a working environment for the enterprise that complements the working environment for the individual offered by Windows, driving enterprise productivity
- **Customer, employee, partner service:** How to build initiatives for employees, customers, and partners using many of the same electronic resources, offering improved service to all at lower cost

Now that you have a general understanding of what the Enterprise Web is, you can start thinking about how you are going to deploy it.

## Deployment Strategies

Before you begin any deployment, you should carefully plan as much as possible, so that your deployment meets the needs of your users and can change and grow with your company.

There are several deployment strategies you might want to consider:

- **Rapid Application Integration Deployment (RAID)**—Use rapid, targeted “seed” deployments to deploy effective, personalized portals for communities. For example, a typical RAID might include the following stages:
  - 2 - 4 week preparation, in which you identify the team, detail your requirements, and assess technical readiness.
  - 5 day Plumtree Services Organization (PSO) on-site, in which you get intensive hands-on training, build your taxonomy, design crawlers and snapshot queries, and transfer community ownership.
  - 30 day follow-up, in which you list outstanding items, assign responsibility, and contact support when necessary.
- **Phased strategy**—Portal deployed in increasingly larger stages. This is the most common and recommended approach. For example, you might include the following phases: pilot, departmental, multi-departmental, global.
- **Enterprise-wide strategy**—Global, enterprise-wide input on design and taxonomy.

## Deployment Best Practices

The following best practices will help you develop a successful Enterprise Web deployment:

- Get executive sponsorship
- Get down into the business
  - Understand the problem, create a business case, identify a business champion
  - Easier information access is not a business need
- Anticipate cultural impacts; nobody likes change
- Invest in planning, but keep in mind that flexibility and adaptation are key
- Treat content as the most important asset
  - Involve content experts
  - Prioritize most important content through alerts and elevation
  - Keep content fresh
- Deploy iteratively: pilots teach and reduce risk
- Emphasize education and training
  - Management first: recruit sponsors and champion
  - Users: explain why, regular training, peer education, online help
  - IT: recognize potential for territorial feelings; recruit allies
- Involve your users: solicit input early
  - Employ pilot groups
  - Enlist corporate training, marketing, help desk
- Invest in marketing: do not assume users will just come
- Develop a long-term plan
  - Survey users, track usage
  - Evolve functionality, prepare governance plan
- Reuse, do not relearn

## Take Advantage of Plumtree Resources

There are many Plumtree resources you can take advantage of when planning your deployment (as well as after you have deployed your Enterprise Web):

- Plumtree Seminars
  - Customer Web seminars and Web seminar archives
  - Inside Track technical Web seminar series
  - Industry Web seminars
  - Monthly customer resource Web seminars
- Plumtree newsletters
  - Bimonthly Deployment Drivers newsletter
  - Weekly Knowledge Base article digest
- Portal adoption tools: in the Support Center at <http://portal.plumtree.com>
- ROI tools: MyROI analysis, self-service workbooks, paper
- Awards submissions: gain recognition
- Regionally-based user groups for: Asia Pacific, Europe, Middle East and Africa, United States - Central, United States - Northeast, United States - Northwest, United States - Pittsburgh, United States - Rocky Mountain, United States - South Central
- Industry-based user groups for: Aerospace, Legal, Public Sector, Navy, Retail
- Support Center: <http://portal.plumtree.com>
  - Knowledge Base
    - Search library of articles by Plumtree engineers, Plumtree Services Organization (PSO)
    - Get up-to-date product documentation
  - Support Services
    - Submit and track open technical support incidents
    - Check technical support policy and worldwide contacts

- Discussions
  - Start and search threads on deployment and portlet development practices
  - Subscribe to e-mail notifications of new threads
- Deployment Drivers
  - Subscribe to Plumtree newsletters and user groups
  - Download portal adoption tools
- Downloads
  - Download product HotFixes
  - Access the Showroom quickly
- Product Roadmap
  - Get answers to Plumtree 5.0 FAQs, check product schedule
  - Submit and track feature requests

Now that you have an idea of how you generally want your deployment to proceed, you need to start thinking about who your users are and what they need.



## Supporting Different Audiences

Your Enterprise Web users might consist of many different audiences—employees, customers, partners—each with different needs and restrictions. You must take into account these different audiences when planning your Enterprise Web solution. Consider the following questions:

- Who will be allowed access to the Enterprise Web—employees, customers, partners?
- What content and services do these audiences need?
- Will users access the Enterprise Web from inside or outside your network?
- Will users access the Enterprise Web from a computer, mobile device, or with some kind of assistive technology (like a screen reader)?
- What roles will you develop internally to administer the Enterprise Web, fill it with content and services, and maintain it?

After you have an idea of who your audiences are and what their needs are, you need to think about how to get the right tools and information to the right people.

### Should You Have Multiple Portals?

There are many benefits of having only one portal:

- Users learn only one navigation scheme
- IT can more easily scale the portal
- It is easy to distribute enterprise-wide communications
- It is easy to instigate enterprise-wide business processes
- Common content management system

In the past you might have had multiple portals for any of the following reasons:

- Different technology: many applications offer portal interfaces
- Different sponsors: many business units want autonomy because they are comfortable with different platforms or devoted to separate brands

- Different audiences: audiences with deep functional needs or with different security—consumers, employees, partners
- Different project schedules: business unit frustration

However, the Plumtree Enterprise Web allows you to avoid multiple portals with the following features, branding flexibility, platform flexibility, separate governance, security domains, and self-service.

If you are starting with multiple portals, there are several ways to take advantage of what you have already done:

- Integrate services from other portals with Plumtree's web services architecture, common portal standards (JSR-168, WSRP), and ability to consume applications written in any modern programming language.
- Security synchronization against common LDAP repository or single sign-on to multiple portals or sites.
- Search allows you to index content in master portal's search engine and federated searches allow you to connect to other content portals' search engines.
- Navigation allows you to link to other portals.

## User Views

You can create users and groups manually, import them from user repositories, invite them by e-mail, or have them create their own user accounts. No matter how you create users, when new users are created, you can control their views of the Enterprise Web through several mechanisms:

- *Default profiles* allow you to define My Pages and other personal settings for new users, determining their initial view of the Enterprise Web. Each new user is based on a default profile.
- *Subportals* allow you to display different branding and features to different audiences. For example, you might create a subportal for a particular customer that includes the customer's logo and company colors and that includes access to My Communities and the Knowledge Directory, but not to My Pages or Administration. Users are stored in folders and the folder determines which subportal each user sees.

- *Communities* allow you to display shared information to specific audiences. Group membership determines which communities users see when they log in to the portal for the first time.
- *Branding* of subportals and communities allows you to display different headers and footers to different users or different community audiences.
- *Security*, which can easily be granted through group membership, allows you to control which objects and features users can access.

## Defining Subportals

Subportals control the overall environment for a group of users—the header and footer; the navigation scheme; the default page displayed at login; access to My Pages, communities, and the Knowledge Directory; and any mandatory links displayed in the navigation.

Every user who accesses your portal sees a subportal. By default, users see the default subportal created at installation. However, you can also create your own subportals to show different interfaces to the different audiences that use your portal. For example, you might want to show one subportal with access to everything to all your employees and another subportal with access to just communities and the Knowledge Directory to your partners.

Subportals are applied to object folders and not directly to users. Users are associated with subportals through the user's parent folder. For example, let's say user U is stored in folder F. If subportal S is applied to folder F, user U and any other user in that folder would view subportal S when they log in. Users must be assigned to a subportal. If users are stored in a folder that does not have a subportal associated with it, those users are automatically associated with the default subportal.



**Note:** If you create a subfolder in a parent folder that includes a subportal association, the subfolder does not automatically default to the parent folder's subportal.

## Defining Communities

Communities are pages shared between the members of a group to collaborate and communicate on a particular project or on departmental goals. For example, you might create communities for any of the following situations:

- Business unit resource centers
- Interactive project workspaces
- Customer or partner management sites
- Dashboards
- Process applications

Communities provide the following benefits to users:

- Consistent user experience and navigation
- Systematic knowledge capture and sharing
- Location of and interaction with experts
- Shortened learning curve and time-to-contribution

When users view a community, they might also see the following features:

- *Subcommunities* are communities that are stored in the parent community's folder. When the user views the parent community, any subcommunities to which the user has access are displayed.
- *Related communities* are all the other communities in the same parent folder as the community. When a user views a community, any related communities to which the user has access are displayed.

You create communities by selecting community and page templates. A *community template* is a collection of page templates. *Page templates* control the layout and mandatory portlets on that page of a community. The templates provide the “minimum requirements” for communities. Community owners cannot remove pages from communities if they are included in the community template, nor can they change the layout or remove portlets included in the page templates; they can only add pages or portlets.

Community owners can control the following features of their communities:

- Unless controlled by the community template, community owners can change the header and footer for a community.
- Portlet preferences allow the community owner to control not only the preferences of the portlet but also whether the portlet header is visible. This is done on a per portlet basis and overrides the settings applied by the portlet creator.
- Community owners can specify whether or not to enable the Community Knowledge Directory, which displays the community members (only users who actively join, not mandatory members) and a community-specific hierarchy created by the community owner.

There are several mechanisms to control how much freedom your users have to create and customize communities:



**Note:** Activity rights are discussed in [“Activity Rights” on page 2-31](#). Access privileges are discussed in [“Access Privileges” on page 2-30](#).

- Activity rights determine whether users can define the templates or create communities:
  - The Create Community Infrastructure activity right allows users to create community and page templates.
  - The Create Community activity right allows users to create communities from those templates.
- Access privileges on the templates determine which templates users can utilize:
  - Users need only Read access to a page template if the page template is part of a community template (to which they must have at least Select access).
  - Users need at least Select access on a community or page template to select those items directly during community creation.
- Access privileges on administrative folders determine where users can create communities. Users need at least Edit access to an administrative folder to be able to create communities in that folder.

- Layout and any portlets defined in a page template determine the basic look of each community page.
- Any pages defined in a community template determine the pages required to be in the community.
- Any headers and/or footers defined in a community template (or defined by the subportal) determine the header and/or footer used in the community.

When you change templates that are already in use by communities:

- If a portlet is added or removed from a page template, that portlet is instantly added or removed on all the communities using that page template.
- Changes to the layout of the page template are applied via a job and might not run instantly depending on the number of jobs in the job queue.

### *Community Development Tips*

- Develop clear implementation plans
- Be flexible for projects that need quick turnaround
- Define a community charter: purpose, audience, key topics
- Identify leadership: community leaders, knowledge stewards
- Consider having all communities developed by cross-departmental project teams:
  - Technical analyst: manages budget, requirements and schedule
  - IT: develops portlets
  - Sponsoring department: serves as content owners
  - User experience team: sets standards for UI, frames
  - Support: creates online help and handles help desk calls
- Ask the following questions:
  - Does it fit the model as a center for collaboration, information sharing?
  - Is there a targeted audience?
  - Would the Knowledge Directory suffice?
  - How often is the content updated?

- The community manager should:
  - monitor discussions, manage community membership, approve content.
  - be the advocate within the business that can establish strong community vision, approve new communities.
- Provide the following training for community managers:
  - Basic: building communities, adding/removing members, basic portlet functionality
  - Intermediate: contributing content, advanced portlets, metadata
  - Advanced: Crawlers and snapshot queries, Excel portlets, managing security
- Have IT create the community, configure security, and then turn over to leadership
- Have leadership drive membership and encourage participation
- Announce new communities:
  - E-mail the community members
  - Advertise with banner ads, news stories
  - Make the community temporarily mandatory

### *Community Management Tips*

If you are managing several communities:

- distribute responsibility by creating folder structures.
- monitor stagnant communities.
- portletize labor-intensive tasks such as adding/removing members.

To make your communities successful:

- keep it fresh: updates, discussions, documents.
- enforce single-source of information.
- avoid portlet page clutter: break down to functional pages.
- form community best practices for your company.
- revisit the community design after getting feedback and monitoring member activity.

Instill a sense of community by:

- publishing photo galleries.
- sharing discussions.

*Community Suggestions*

Here are few ideas for how to use communities:

- Business Unit Resource Center (Line of Business Communities)

<b>Audience</b>	<ul style="list-style-type: none"><li>• Business unit or department</li><li>• Customers of that business unit or department</li></ul>
<b>What to put in it</b>	<ul style="list-style-type: none"><li>• Community documents, links, calendar</li><li>• Metrics</li><li>• Expert finder</li><li>• Q&amp;A</li></ul>
<b>Success indicators</b>	<ul style="list-style-type: none"><li>• Strong departmental or group identity</li><li>• Existing intranet as content source</li><li>• Motivated community owner</li></ul>
<b>Pitfalls to avoid</b>	<ul style="list-style-type: none"><li>• Static page that people visit and forget</li></ul>
<b>Suggested Plumtree tools</b>	<ul style="list-style-type: none"><li>• Excel Framework</li><li>• Plumtree Content Server</li></ul>



- Interactive Workspace (Collaborative Communities)

<b>Audience</b>	<ul style="list-style-type: none"> <li>• Ad hoc or established project workgroups</li> </ul>
<b>What to put in it</b>	<ul style="list-style-type: none"> <li>• Project task list</li> <li>• Document management and archive</li> <li>• Project calendar</li> <li>• Threaded discussions</li> <li>• Project metrics</li> </ul>
<b>Success indicators</b>	<ul style="list-style-type: none"> <li>• Members spread out</li> <li>• Project has specific objectives and milestones</li> <li>• Project has outgrown e-mail and file-shares</li> </ul>
<b>Pitfalls to avoid</b>	<ul style="list-style-type: none"> <li>• Dustbin of history: old projects, communities that do not go away</li> <li>• Ghost town: two or three people are probably not enough</li> </ul>
<b>Suggested Plumtree tools</b>	<ul style="list-style-type: none"> <li>• Plumtree Collaboration Server</li> <li>• Plumtree Studio Server</li> <li>• Excel Framework</li> </ul>

- Customer or Partner Management Site (Sales and Service-Oriented Communities)

<b>Audience</b>	<ul style="list-style-type: none"><li>• Customers or partners</li></ul>
<b>What to put in it</b>	<ul style="list-style-type: none"><li>• Key customer or partner resources: documents, calendar</li><li>• Self-service access to CRM or PRM system</li><li>• Feedback mechanism</li><li>• Customer-to-customer or partner-to-partner: facilitate community</li></ul>
<b>Success indicators</b>	<ul style="list-style-type: none"><li>• Portal-only access for critical information</li><li>• Responsiveness to customer/partner feedback</li></ul>
<b>Pitfalls to avoid</b>	<ul style="list-style-type: none"><li>• No human input</li></ul>
<b>Suggested Plumtree tools</b>	<ul style="list-style-type: none"><li>• Plumtree Collaboration Server</li><li>• Plumtree Studio Server</li><li>• Excel Framework</li></ul>

- Dashboards (Analytic Communities)

<b>Audience</b>	<ul style="list-style-type: none"><li>• Management</li></ul>
<b>What to put in it</b>	<ul style="list-style-type: none"><li>• Performance metrics</li><li>• Financial documents</li></ul>
<b>Success indicators</b>	<ul style="list-style-type: none"><li>• Support to enforce consistent data formatting</li><li>• Portal-only access for critical information</li><li>• Culture of accountability based on metrics</li></ul>
<b>Pitfalls to avoid</b>	<ul style="list-style-type: none"><li>• Make sure the dashboards have fresh data</li><li>• Make sure security works appropriately</li></ul>
<b>Suggested Plumtree tools</b>	<ul style="list-style-type: none"><li>• Excel Framework</li><li>• Enterprise Class Portlets for SAP and PeopleSoft</li><li>• Certified Portlet Suite for Cognos</li></ul>

- Business Process Applications (Process Communities)

<b>Audience</b>	<ul style="list-style-type: none"><li>• Users involved in process</li></ul>
<b>What to put in it</b>	<ul style="list-style-type: none"><li>• Published content</li><li>• Data from multiple systems</li><li>• Workflow</li><li>• Metrics</li></ul>
<b>Success indicators</b>	<ul style="list-style-type: none"><li>• Simple navigation, consistent branding a priority</li><li>• Unified search criteria</li><li>• Looking to utilize reusable components, common foundation</li></ul>
<b>Pitfalls to avoid</b>	<ul style="list-style-type: none"><li>• If you do not have a process, the software will not do it for you</li></ul>
<b>Suggested Plumtree tools</b>	<ul style="list-style-type: none"><li>• Plumtree Content Server</li><li>• Collaboration Server</li><li>• Plumtree Search</li><li>• Plumtree Studio Server</li><li>• Excel Framework</li></ul>

## Branding Subportals and Communities

You can apply different headers and footers to different subportals and communities. Subportal headers and footers are applied to the entire portal except for communities that have their own custom headers and footers. Headers and footers are just special portlets that can be used only in subportals and communities for branding purposes.



**Note:** The header affects not only the banner image, but also the style used by the page (because it contains the style sheet reference).

When you import the branding object package (the .pte file), some header and footer templates are automatically created, which you can then easily customize. Users must be in the Branding group (created by the branding package) to create headers or footers or modify them from communities.

There are a few things you should know about branding creation, modification, and use:

- Headers and footers can only be created from the administrative hierarchy.
- The same header or footer portlet can be used by different communities but can look different in each community by changing the properties of the portlet.
- Headers and footers can be modified directly from the community. Users in the Branding group can click the header or footer name to edit the portlet properties and change the look of the header or footer.
- You cannot modify headers and footers from a subportal. If you use a header or footer that is used by multiple communities, each having their own modifications, the subportal uses the first set of specifications entered for that header or footer.

### *Branding Tips*

- For typical community owners, it is much easier for them to modify existing headers and footers by uploading images and entering property values rather than writing their own HTML. This is especially true for customizable banners that require Plumtree Content Server tags.

- Since community owners cannot modify the style (css) used by a header, specify the style in the name of the header to make sure that the community owner makes suitable modifications to the header. For example, if the header uses an orange-based style, the community owner should not add green images (this depends on taste of course). Since only the name of the header is displayed, the only way that the community owner knows what style the header uses, without actually applying the style, is to see it in the name.
- You do not need to use the branding templates to make headers and footers. They are provided to give you an encapsulated mechanism for creating, editing, and publishing Web properties that can be used as headers and footers. You can create a Web page using any HTML editor, save it to a Web server, point a portlet web service to it, and specify that the resulting portlet is a header. You can also use HTML editors and the branding templates in conjunction. You can create HTML in the editor and copy that code in the proper steps of the header wizard in the branding templates.
- The portal style (css) is typically set using the branding templates. If you bypass the branding templates, you can hardcode the style in the HTML. Otherwise the portal uses the last available style. For example, if the subportal header specifies a style but the community header does not, the portal uses the subportal header style even when it displays the community header.

## Controlling User Views with Access Privileges

If you want to support multiple, discrete audiences, in which the different audiences should never see other groups' objects, you should create a top-level folder and do the exact same thing within that folder that you would do for a single portal. Secure the folders so that only users in the proper audience can access objects in that folder. Keep a central administrative resources folder for shared objects. For example, your company might have multiple subdivisions. Subdivisions should not access other subdivisions, but your corporate office should be able to access all divisions for oversight.

Now that you know who your users are and what they need, you need to think about how to secure everything and how to manage objects and content.

## Securing and Managing the Enterprise Web

When considering management strategies for your Enterprise Web, there are a few things you need to think about—who will have access to what, who will perform what duties (what roles need to exist), and how much control you want to give to each role. There are several features that help you implement your strategy:

- Object security is provided through Access Control Lists (ACLs) on each object. These lists define which users and groups can view or modify the objects. For example, a user must have access to a portlet to be able to see the portlet on a community page.
- Feature security is provided through activity rights assigned to groups. For example, a user must have the right to access administration to perform any administrative tasks in the portal.
- Nested groups allow you to create roles. You can create roles by creating groups specifically for assigning activity rights and access privileges. You can then add users or other user groups to the role groups to assign roles. For more information on roles, refer to [“Defining Roles” on page 2-32](#).
- You can create development, testing, and production deployments and use object migration to move objects from one deployment to another. This allows you to strictly control who has the ability to create new objects, to test object security and process load, and to make objects available to users only when they are working as planned.
- You can use page templates and community templates to control the look and functionality of communities. For example, you might require each department to include departmental contact information in their communities.

Start by thinking about portal security. Portal security is based on two things:

- *access privileges* (object security), which determine who has access to what.
- *activity rights* (feature security), which determine who can perform what actions.

## Access Privileges

Your end-users and low-level administrators should generally be able to perform their jobs just by being assigned access privileges to portal objects. Without any special activity rights, users can view and/or manage the following objects just by having the appropriate access privileges:

- documents and Knowledge Directory folders (available through banner search and the Knowledge Directory)
- users (available through banner search)
- communities (available through banner search and the Join Communities page)
- portlets (available through banner search, communities, and the Add Portlets page)



**Note:** Users with access to objects other than those listed above will not be able to view or manage those other objects unless they also have the Access Administration activity right.

The following access privileges are available:

- Read allows users to view an object.



**Important:** Users need at least Read access to both the object and the object's parent folder in order to browse to the object or to have that object appear in searches. However, users can see an object that they have access to if it is presented to them. For example, if users have access to a portlet, but not to the portlet's parent folder, they will still be able to see that portlet if it is placed on a community to which they have access.

- Select allows users to choose an object for use. In general, you want to give users Select access on objects. For example, you must give users Select access to a portlet if you want them to be able to add that portlet to their My Pages, and you must give users Select access to a community if you want them to be able to join that community.
- Edit allows users all rights except deleting the object and editing the object ACL.
- Admin allows users full rights on an object.



## Mandatory Administrators

Groups and users with Admin rights on a folder are mandatory administrators of all child objects and folders. This allows the creation of domain administrators (administrators of a group of folders).

## Special Cases

Certain objects have special requirements. The following objects require the Everyone group to have Read access:

- Authentication Sources
- Filters
- Document Types
- Properties

One other special case is that users do not have ACLs applied to them directly. They inherit the ACL of their parent folder. This is because there are generally so many user objects that managing by user would be cumbersome.

## Activity Rights

Most activity rights have to do with creating objects or accessing areas of the portal, but users can also be given the right to delegate their own activity rights to other groups, and you can create custom activity rights.

The most important thing to remember about activity rights is that users do not need creation rights to manage objects; they just need the proper access privileges to the object itself and a way to access the object (via browse or search). If you want to make sure that certain users do not have access to modify any administrative objects regardless of their access privileges to those objects, do not give those users the Access Administration activity right. Without this right they cannot browse the administrative hierarchy or search for most administrative objects, but they might still be able to modify portlets, communities, and users because these objects are searchable through banner search and therefore users only need the proper access privileges to be able to modify these objects.

## Defining Roles

Roles are just groups specifically created for the assignment of activity rights or access privileges. You can use these roles in combination to provide the correct security for your users and user groups.

You probably want to create groups specifically for the purpose of assigning portal activity rights, without including any access privileges. Generally, roles that include activity rights are held by different people in each department (for example, each department probably has its own community administrator). Keeping the access privileges separate from the activity rights allows you to define activity roles once and then apply them throughout your company, instead of having to define the same set of activity rights for each department. For example, if you want to create a content administrator role, you can create a group with the following activity rights: Access Administration, Create Crawler, Create Job, Create Filter, Create Document Type, and Access Utilities.

You can usually assign access privileges to your existing user groups, because access is generally granted along departmental lines, and, more than likely, your user groups are also along departmental lines. For example, you might give the Marketing group Select access on the Marketing folder and all Marketing objects. Depending on the size of your department, you might also need to create roles for Admin and/or Edit access; if there are only a few people who need Admin or Edit access to an area of the portal, you might want to just change the access for those individual users. You do not need to give creation activity rights (such as Create Portlets) to every role; you only need Edit access on an object to edit the object. Therefore, you create roles that can manage existing objects but cannot create anything.

Put the most powerful groups at the bottom (the deepest level) of a group hierarchy. Because groups inherit the rights of the parent groups, the groups with most rights are the groups furthest down the group hierarchy. For example, the IT Managers group should be a child group of the IT group, not the other way around. This is especially true of groups with activity rights directly assigned to them. They should be as low in the hierarchy as possible.

### Common Activity Rights Roles

Here are a few suggestions for common roles used to assign sets of activity rights:

*Table 2-1: Common Activity Rights Roles (Sheet 1 of 2)*

Role	Suggested Activity Rights
Content/Document Administrator	<ul style="list-style-type: none"><li>• Access Administration – to access the administration hierarchy</li><li>• Edit Knowledge Directory – to create new document folders</li><li>• Create Crawlers – to create new crawlers</li><li>• Create Data Sources – to access secured documents</li><li>• Create Document Types – to force metadata onto documents</li><li>• Create Filters – to automatically manage folders</li><li>• Create Jobs – to run jobs</li><li>• Access Utilities – to approve documents</li><li>• Access Smart Sort – to re-sort entire folders of already categorized documents</li></ul>
Community Creator	<ul style="list-style-type: none"><li>• Access Administration</li><li>• Create Communities – to create communities</li><li>• Create Community Infrastructure – to create community and page templates</li></ul>
Portlet Creator	<ul style="list-style-type: none"><li>• Access Administration</li><li>• Create Portlets – to create portlets</li><li>• Create Web Service Infrastructure – to create the remote server and web service to create truly custom portlets</li></ul>

Table 2-1: Common Activity Rights Roles (Sheet 2 of 2)

Role	Suggested Activity Rights
Group/User Creator	<ul style="list-style-type: none"><li>• Access Administration</li><li>• Create Admin Folders – to make new admin folders to store users</li><li>• Create Subportals – to modify the user experience of users</li><li>• Access Utilities – to create default profiles to apply initial layouts to users</li><li>• Create Authentication Sources – to create authentication sources</li><li>• Create Jobs</li><li>• Create Profile Sources – to apply user information to synchronized users</li><li>• Create Groups – to create groups</li><li>• Create Users – to create users</li><li>• Delegate Rights – to delegate rights to users (create activity groups)</li></ul>

## Defining an Administrative Object Hierarchy

The Administrative Object Directory is similar to the Knowledge Directory; it is a hierarchical folder structure (up to 10 levels deep), but it stores administrative objects rather than files. The folders can store any type of administrative object (for example, crawlers, portlets, or users). Within each folder, objects are automatically grouped by object type to ease management. Each folder is secured, and objects created in that folder default to the ACL of the parent folder.

Consider the following tips when creating your administrative object hierarchy:

- Start with the end-user hierarchy. Although you can create a hierarchy based on the organizational or management structure, this is often not the best organization for end-users (users who browse the portal rather than manage parts of the portal). End-users can see the administrative hierarchy in a few places in the portal. For example, by default, the Add Portlets and Join Communities pages search the administrative hierarchy for available portlets and communities and try to display a list of objects without showing their parent folders. However, users can choose to browse the hierarchy.

The best thing to do is to start by creating the hierarchy for only communities and portlets (including portlet bundles) and hiding the administrative objects created during install. For example, you might want to move all objects meant for administrators to a particular folder and restrict access to the folder so that end-users will not see it if they browse the hierarchy.

The organization of the objects meant for administrators can be based on administrative structure or by object type or by topic.

- Hide folders that are intended for administrators (as mentioned in the previous bullet).
- Organize objects by topic rather than by object type (objects are automatically grouped by type within each folder).
- Preset folder ACLs. If your group hierarchy is fairly stable, preset the ACL on Admin folders with both groups and their subgroups. This takes some planning but it will be of great use to portal managers. Since it is easier to remove than add groups and users to the ACL of objects, pre-add as many groups to the ACL of objects folders. When an object is created in that folder, it defaults to the ACL of the folder. If the object is to be

restricted to a subset of the folder, the owner of the object can then remove groups from the object. For example, assume all the members of the IT Managers group are members of the IT group. It would be helpful to add both the IT Managers and the general IT group to the ACL of the IT folder. For objects that are meant for just IT Managers, simply remove the general IT group. If only the general IT group was initially added to the parent folder, you would have to remove the general IT group from the object and search for the IT Managers group to add it to the object. If you are constantly adding groups, this might become cumbersome.

- Do not rely entirely on folder ACLs. Do not always rely on folder ACLs to control access to objects. Remember, technically a user only needs Read access to a portlet if that portlet is already on that user's My Page or community. It is better to set security on the object than the folder.
- Always manage by groups. It is always easier to manage by groups than users, especially because you can put groups in groups. For example, assume IT manages 100 objects. user U is the IT manager. If only user U was added to the ACL of objects then it would be very difficult to also let user X manage IT objects since user X would have to be added individually to all IT objects. If the IT Manager group is added to the ACL of the IT objects, then any user added to that group would inherit the rights of that group.
- Start from the top. When you are manipulating folder security for a lot of folders, you should always start from the top. Remember that propagation goes from the top down. Therefore if you change the ACL of a subfolder and then change the ACL of the parent folder and select propagation, the changes you made to the subfolder will be lost. Since propagation is handled by jobs that are automatically created in the Intrinsic Operations folder, you will not see the changes immediately, but the jobs will run and the proper security will be set.

Now that you have developed a security and management strategy, you can think about what content you want to make available through the Enterprise Web.

## Providing Access to Divergent Sources of Content

Where does the information you hope to bring into the Enterprise Web reside? Is it located in a database, a special file format, or a secure environment? The answers to these questions will help you determine how to set up and define the different tools that allow you to make content available through the Enterprise Web.

### Types of Data

Data is anything in the enterprise, either structured or unstructured. The most common sort of unstructured data is in the form of documents. Documents can be stored in any system. Plumtree pulls documents into the Enterprise Web through various mechanisms, including crawlers, searches, and portlets. When documents are crawled, they are indexed in the Knowledge Directory. All common file types can be full-text indexed, but Plumtree has extra capabilities to extract metadata from the following types of documents:

- Windows Files
- HTML
- Microsoft Office
- Lotus Notes
- Microsoft Exchange
- Adobe PDF
- Documentum
- Novell



**Note:** This is not a complete list of the formats supported. Some format support, such as Microsoft Exchange, is supplied as an individual component of the Enterprise Web. Also, through the Enterprise Web Development kit, you can expand the types of documents and systems from which you can extract metadata for categorization in the Knowledge Directory.

## Location of Data

As mentioned previously, data can be stored in any system, for example: file repositories, ERP systems, databases and so on.



**Note:** Through the Enterprise Web Development kit, you can create and customize crawlers. For example, you might create a FileNET crawler, customize the Windows File System crawler to extract additional metadata from documents, or crawl a custom database.

## File Repositories

- Microsoft Windows Files

The most common source of information is files located on a Microsoft Windows network. A file crawler attempts to import content from each document it finds. File crawlers start crawling at one file folder of any UNC-compatible directory (`\\<computer_name>\<folder_name>\<subfolder_name>`) and continue crawling into that folder's subfolders. Things to consider about Windows network system files include:

- Is there information in a secure folder that will require authentication to access? What domain and user name can be used to gain access?
- Do you want to import Windows NT security information with the document or rely on Plumtree security?

- HTML—World Wide Web

Web crawlers start crawling at a single Web page and continue to follow links to connected pages. You can control the content that enters your Knowledge Directory by using filters and special Web crawler options. The important questions to ask about Web information are:

- Is there a proxy server? What is the configuration of the proxy server?
- What is the target site security?



- Microsoft Office
  - Are the Office documents in a shared folder that can be accessed through a UNC-path to that folder?
  - Are the Office documents attachments to e-mail?
  - What domain and user name can be used to gain access?
- Lotus Notes File Structure
  - Do you want to import security along with the Notes documents?
  - What is the location of the Notes databases?
  - What type of access is required for the databases to be crawled?
- E-mail

E-mail is everywhere and is a critical source of information that most people need to do their jobs. However, in most organizations, e-mail information stays with the user. With Plumtree, however, you can crawl information from Microsoft Exchange or Lotus Notes and have it be searchable from the portal. Considerations to think about include:

  - Are there public folders set up on the mail system?
  - What type of access will be required?
- Adobe PDF
- Documentum
- Novell

## Discovering Content

You should create a plan to discover the content available in your company and to decide how to handle that content in the Enterprise Web. For example, you might want to perform the following actions:

1. For each functional area, create a focus group made up of information managers:
  - Content managers, authors
  - Librarians
  - Subject matter experts
  - Application managers
  - Database managers
2. At the focus group meetings:
  - Present a 2-hour hands-on introduction to the portal. This is vitally important for users to understand.
  - Hand out a content inventory worksheet with instructions on how to identify knowledge assets like:
    - Web sites
    - Databases
    - File servers
    - Document management systems
    - People

Have attendees post completed worksheets online or send via e-mail.
3. Analyze the aggregated results of the content inventory and make decisions on what content should be included in the Enterprise Web. Then create and document policies to track decisions, for example, how to handle certain document types.

Analyze the following information about the knowledge assets:

- Metadata
- Security
- File types
- Document types
- Data sources

Analyze the following business processes:

- Document staging and approval
- Electronic publishing guidelines
- Version control
- Quality control
- Distribution requirements, tactics
- Maintenance procedures
- Archiving

4. Keep the inventory as a content management history log.

After determining what content you want to include in the Enterprise Web, you need to develop a taxonomy.

## Defining a Knowledge Directory Taxonomy

There are several approaches to developing a Knowledge Directory taxonomy:

- Manual/Top-down: Individual-driven; can entrench obsolete or arbitrary categories
  1. Conduct knowledge gathering
  2. Work with functionally- or subject-based individuals or focus groups to identify major categories of interest; subdivide as necessary to build taxonomy
  3. Categorize content into this taxonomy

- Automated/Bottom-up: Content-driven; can reveal new associations of information
  1. Conduct knowledge gathering
  2. Identify major document collections; analyze content to reveal major and minor topics of information
  3. Build taxonomy based on the relationship of these topics; can be automated using textual analysis tools

No matter what approach you take, you need to decide on a taxonomy classification scheme. You can use the same scheme throughout your entire taxonomy, or a different scheme for the different branches of your taxonomy. Table 2-2 describes some possible schemes; they are listed from hardest to easiest to implement.

Table 2-2: Taxonomy Classification Schemes

Method	Definition	Examples
Subject-oriented	Information categorized by subject or topic	water pollution, soil pollution, air pollution
Functional	Information categorized by the process to which it relates	employment, staffing, training
Organizational	Information categorized by corporate departments or business entities	HR, Marketing, Accounting
Document Type	Information categorized by the type of document	presentations, expense reports, press releases
Other Document Attribute	Information categorized by a specific metadata field or document attribute	author, data source, creation date

After deciding on your approach and scheme, you are ready to create your taxonomy plan. Keep the following best practices in mind:

- Keep your audience in mind
  - Recognize that users might think about and look for information in different ways
  - Consider multiple taxonomies for disparate audiences
  - Use familiar vocabulary and organizational schemes to ensure a logical browsing experience
  - Keep in mind that browsing and searching are both critical to knowledge discovery
- Strive for a subject-based categorization approach
  - It is the most objective, consistent, intuitive, extensible approach
  - It is the most compatible with auto-categorization tools
  - It might require cultural change; corporate users tend to think along organizational, functional, or document type lines
- Be consistent
  - Try to use a single classification approach; if it makes sense to combine approaches, then keep it consistent on the peer level
  - Use consistent vocabulary, thesauri
  - Maintain consistent degree of generality in sibling categories
- Control depth
  - A flat taxonomy ensures that users can find information quickly with fewer clicks
  - Guideline: 3-6 levels deep
- Control breadth
  - A focused taxonomy ensures that users can easily “digest” the scope of information
  - Guideline: 10-15 top-level categories

## Managing Content

There are several solutions available for managing content in the Enterprise Web:

- You can manage content in a portal publishing repository to:
  - extract desired, portal-ready content from disorganized hard drives and file shares also containing old, draft, or inappropriate content.
  - enable centralized control of portal content by designated content administrators.
  - allow for reorganization of content and facilitate the use of mirror crawlers.
  - focus document and metadata management at the publishing stage.
- You can manage content in existing, distributed repositories to:
  - avoid any content migration effort.
  - focus document and metadata management at the portal's document approval stage.



**Note:** This solution is necessary if you are crawling content you do not control.

- You can integrate the portal with a document management system to:
  - focus document and metadata management at the repository submission stage.
  - enable document management from the portal via portlets.
- You can create new content through Plumtree Content Server, which includes:
  - data entry templates that make it easy for your business audience to contribute content in a consistent way.
  - workflow that assures that documents will go through the appropriate stages, for example, writing, editing, approval, and publishing.
  - presentation templates that display content in a consistent way.
  - library services, to include a wide range of sites beyond the portal
  - publishing to portal pages, the Knowledge Directory, intranets, or extranets.

## Defining Search

Plumtree Search, an integrated part of the Plumtree Enterprise Web Suite, allows users to quickly and efficiently find a wide variety of information from sources across the enterprise, both inside and outside the Plumtree Corporate Portal and related Content Server and Collaboration Server products. Plumtree Search works with Plumtree's Directories and Web Services infrastructure to help employees do their jobs. Salespeople can find contract resources needed to close deals; marketing executives can find in-progress design documents for new products; customer service representatives can find resources stored in a variety of CRM, file, and Web repositories.

There are a number of possible sources of searchable content in the Enterprise Web Suite, and it is important to understand the options for providing that content to end-users:

- **Knowledge Directory:** The core of the Plumtree knowledge management infrastructure is the Knowledge Directory—a hierarchy of folders that contain links to files of various formats, stored in different types of repositories. Files can be crawled into the Directory or manually submitted, and can be filtered into the folder hierarchy (also known as a taxonomy) in order to provide an entry point to high-quality, organized content. In addition to the out-of-the-box functionality, virtually any repository can be made searchable through the creation of crawler web services. All items in the Knowledge Directory can be searchable.
- **Collaboration Server:** The project workspaces provided by Collaboration Server contain documents, threaded discussions, announcements, and task lists contributed and managed by distributed teams. All items in Collaboration Server can be searchable.
- **Content Server:** The form-based data entry and file management provided by Content Server allows specialized content submitted by users to be published and surfaced in the portal through portlets. All published content items associated with portlets can be searchable.
- **Portal Administrative Objects:** Users, web services, portlets, crawlers—all the objects that make up the administrative infrastructure of a portal are searchable. End-users can search for users (to view profile and expertise information), communities (to visit or join), and portlets (to add to a My Page). Administrators (who need to create

and manipulate all types of objects) can search for a wider variety of items and have more advanced options in their search results.

- **Non-portal Searchable Content:** Legacy search engines and repositories with pre-existing search or query functionality can often contain valuable sources of content that for various reasons cannot be crawled into the portal or managed through Collaboration Server or Content Server. With search web services, any repository that can respond to queries can be extended with a web services adapter so that it can be searched from the portal. Results from a number of disparate search providers (both inside the enterprise and on the internet) can be aggregated in this way.

## Crawler Web Services Versus Search Web Services

Another way to categorize the types of searchable content in a Plumtree Enterprise Web deployment is based on where the original content resides:

- **Plumtree-managed data:** For portal administrative objects, Collaboration Server items, and Content Server items, the original representation of the searchable items resides within the Plumtree system; no external repository is required. For example, the original data underlying a user is present in various tables in the portal database; the PDF file that is part of a Collaboration Server project is present in the Plumtree Document Repository. The link and metadata information for this kind of data comes from crawler web services.



**Note:** The Plumtree Document Repository is a low-level service used by the Plumtree Corporate Portal, Collaboration Server, and Content Server to store and retrieve files in a platform-independent way. It has no end-user facing component or API; it is a behind the scenes utility service for products in the Enterprise Web Suite.

- **Linked data:** The Knowledge Directory stores links to crawled or manually submitted content that resides in other repositories, providing a unified and organized view without having to maintain a redundant copy of the data. For example, when a PDF file is crawled into the Directory from a Windows file system, the file remains in its original location in the file system. A link to the file is presented in the Directory (and stored in



the portal database), and text and metadata are extracted from the file for indexing in the Plumtree Search Server. Similarly, if a file is manually submitted from a Documentum repository, the Knowledge Directory merely contains a link to the file in Documentum.

- **Completely external data:** For data in a third-party search engine or a repository with embedded search capabilities (for example, a Siebel CRM system), not only is the original data managed outside Plumtree, but there is also no link to the data in the Knowledge Directory. This might describe Web data that has been indexed by Google on the intranet or personnel information which must remain in a restricted-access human resources system of record. Search results for this kind of data come from search web services.

With this backdrop, we can state that:

Crawler web services (CWSs) allow you to extend the types of linked data crawled into the Knowledge Directory. Implementing a CWS means using the Plumtree Enterprise Web Development Kit and the API of the target repository to create a web service. That web service can respond to a few types of requests: “Give me all the documents in a folder”, “Give me all the metadata in a document”, and “Give me a copy of the file for a document (so that I can extract the text)”. The portal can then invoke this web service to crawl the items into the Directory, where it can be searched with no additional effort.



**Note:** There is actually nothing requiring the repository data to be a file. For example, a crawler web service can be used to crawl structured records from a database. In that case, there may be no file associated with an item, only metadata.

A crawler web service should be used when:

- There is an API for the target repository that can be used to retrieve folders, files or records, and associated metadata for the purposes of indexing.
- The repository contents change on a timescale of hours and days, rather than minutes and seconds. The crawler (and associated agents) that use the web service to index content will run periodically, so rapidly churning repositories will lead to mismatch between the searchable indexed content and the actual original data.

- There is a requirement for presenting an organized view of the (links to) data as part of a central browse hierarchy. Crawling links into the Knowledge Directory can provide such a view.
- There is a requirement for taking advantage of advanced search features provided by the Plumtree Search Server, such as metadata search, best bets, metadata weighting, and thesaurus substitution.

Search web services (SWSs) allow you to offer external search functionality to users, leveraging pre-existing investments in search engines and search-enabled repositories. Implementing an SWS means using the Plumtree Enterprise Web Development Kit and the API of the target search provider to create a web service. That web service can respond to a request “Give me items matching the search string, augmented by a username and password to apply relevant security”. The portal can then issue a search string and authentication information to this web service in parallel with other search web services to receive results from each. The results can be presented alongside results for the same search string against Knowledge Directory items linked in the portal, interleaving the results from each source. The key point to realize is that search web services rely on the native search capabilities of a target search provider, be they database or index based.

A search web service should be used when:

- There is no API available to allow crawling of content, or it is impossible or undesirable to crawl content into the Directory for reasons of security, scale, or costs invested in pre-existing search technology.
- Up to the second or up to the minute search result accuracy is required, so the latency introduced by periodic (typically daily) crawls is unacceptable.
- There is no requirement or need to browse the content in a hierarchy, only search results are needed.
- There is no need to take advantage of advanced or vendor-specific search features from the various providers of search web services that will be aggregated in the portal. In order to issue a simultaneous search across a wide variety of web service providers, SWS in Plumtree 5.0.x uses a “lowest common denominator” protocol which sends little more than the search string and the user authentication information to each provider.

## Understanding What Users Are Searching For

Plumtree Search provides highly relevant results to keyword queries, taking into account word order and phrases, language-specific analysis of alternate word forms, and the meta-data structure of searchable items, among other things. Plumtree Search also gives users effective cues as to the relevance of search results through keyword highlighting. It is still important to help steer users to the most relevant documents, through planning the type and organization of portal content and by exploiting features of Plumtree Search that allow administrative control over search results. Before you can exert control over search results, you must have some insight into what users are searching for.

Plumtree 5.0 introduces a Log Report feature that can be used to summarize search usage over periods of days, weeks, and months. The reports show the most frequent searches performed by portal users, and they show the most frequent searches that returned zero results. The reports can be configured to run periodically as an external operation job on the Automation Server. The default job is configured to examine 7 days worth of logs and to run weekly, but additional jobs can be configured to summarize the search logs for different periods of time as well (for example, daily or bimonthly). The reports are stored in the portal database, and old reports can be deleted through the administrative interface.



**Note:** In 5.0 and 5.0.1, Log Reports are accessible through the Search Server Manager utility in the Administration area. Only users in the Administrators group can access this utility, preventing content maintainers who are not IT personnel from viewing the log reports. In 5.0.2, Log Reports and related utilities have been separated into a Search Results Manager utility which is secured with a new Activity Right. Content maintainers can be given access to Log Reports and related functionality without having to give them full portal access.

Log reports can provide actionable information—armed with the most frequent queries, a content maintainer can edit folder structure, metadata, community links, community announcements, and more in order to streamline access to the most requested content. Frequent searches that return no results can be even more useful—they indicate information that users are looking for but not finding. The *Administrator Guide for the Plumtree Corporate Portal* and Plumtree Knowledge Base article DA\_131\_654: “5.0 Search Server Log Reports” provide technical details about how to set up and access log reports.

The main limitation of 5.0 Log Reports is that the reports are generated from the complete search logs for a given day or month and are not associated with particular users. So there is no way, for example, to determine what members of a specific community or subportal are searching for, as distinct from the complete portal user population. There are a number of more sophisticated usage tracking options available however, most notably PTracker, which uses information from Web Server logs and the portal database to provide detailed reporting capabilities about search and a host of other types of portal usage.

The next section describes additional Plumtree Search features that can be used to more directly influence search results and guide users to the most relevant content.

## Influencing What Users Find

*Best bets* allow administrators to control the top results returned for specific search terms for portal banner search. When a user's banner search query exactly matches a best bet search phrase (except for case or spacing), the best bet results configured by the administrator display first in the relevance ranked result list. The phrase "Best Bet" displays next to each result to inform users that the result has been judged especially relevant to the query.

An administrator creates a best bet by defining a search query (for example, "enterprise web") known as the best bet trigger. This corresponds to a search that users are expected to submit, and can be derived from lists of frequent searches as obtained from log reports, PTracker, or similar usage tracking systems. The administrator can then choose up to three results to be associated with the trigger (for example, the 2003 memo from the CEO on the Enterprise Web market), and can order the results as desired. Up to thirty best bet triggers can be created, each mapping to a maximum of three results.

If portal users are submitting searches that return no results, a best bet can be created to associate their common search terms with the appropriate content. Similarly, if users are submitting searches that return a large number of documents that are equally relevant on the basis of text and metadata alone, a best bet can be created to ensure that specific results come out on top.

## Related Materials

- The *Administrator Guide for the Plumtree Corporate Portal* includes:
  - technical details of how to organize a Knowledge Directory and set up the administrative infrastructure needed to fill it with content.
  - how to create snapshot queries and the associated content snapshot portlets for displaying the results of a stored search in a portlet.
  - technical details of how to set up federated searches in which a portal can be used to search content in other portals or (more typically) in non-portal repositories that can provide search web services. The *Enterprise Web Development Guide* has information on how to actually build search web services for non-portal repositories.
  - information on operation and maintenance of the Search Server and associated portal agents which index and refresh portal searchable content.
  - technical information about a number of search features - log reports, best bets, thesaurus, and search result grouping based on metadata.
- Plumtree Knowledge Base article DA\_131654: “5.0 Search Server Log Reports”
- Plumtree Knowledge Base article DA\_130470: “Best Bets Feature in 5.0 Search”
- Plumtree Knowledge Base article DA\_138009: “Using the Search Server Thesaurus”
- Plumtree Knowledge Base article DA\_131637: “Customizing 'Sort By' Categories for 5.0 Search”

## Providing Interactive and Informational Applications to Users Through Portlets

Portlets are applications that are embedded in a portal and can be interactive or merely informational. They are able to communicate preferences with the portal and communicate with other portlets. A portlet is not merely a web service. A portlet can be deployed as an application running on the same platform as the portal, a remote application, a frame, or a browser control.

There are several components involved in a portlet. A portlet must ultimately be based on a web service. The web service controls the bulk of the portlet settings, for example, the URL and cache settings. The portlet controls the name, width, type, and administrative preferences (if available). You can also create portlet templates allowing you to create multiple instances of the same portlet, with each instance looking different or displaying different information based on the administrative preferences set in that instance.

There are many portlets available with Enterprise Web components, but you can also create your own portlets through Plumtree Studio Server or Plumtree Content Server, or write your own portlets using the Enterprise Web Development Kit (EDK).

### Why Build Portlets?

There are many reasons you might want to build your own portlets:

- You can simplify user experience by avoiding separate logins for each service, avoiding the complexity of exposing users to a complete application, or you can customize and condense application experience.
- You can Web-enable proprietary systems to avoid Web site sprawl, avoid security and user interface costs, and provide Internet access through the Enterprise Web gateway.
- You can draw users to a broad experience like quality and safety checks or benefits enrollment.



**Important:** Do not Web-enable expensive systems. Do not create portlets that require a lot of maintenance, but do not provide big cost or time savings.

## Which Portlets to Build First

Consider the following questions when deciding which portlets to build first:

- What do people want?
- What is a difficult or time-consuming to do?
- What do people use regularly?
- What is important to people's jobs?
- What drives the business?
- Which systems are proprietary?

Here are some ideas for portlets:

- |                  |                         |                                 |
|------------------|-------------------------|---------------------------------|
| • Lunch          | • Sales support         | • WYSIWIG publishing            |
| • E-mail         | • Invoice lookup        | • Time zone calculators         |
| • Pager          | • Time and expense      | • Process-based portlets        |
| • Calendar       | • Work order status     | • Project artifact dashboard    |
| • ERP access     | • Employee directory    | • Business initiatives calendar |
| • Yellow pages   | • Executive dashboard   | • Conference room scheduling    |
| • Sales analysis | • Simplified navigation |                                 |

## Building Good Portlets

For detailed information on building portlets, refer to the *Enterprise Web Development Guide*.

Here are some things to keep in mind when building portlets:

- Keep it simple
- Support drill-down
- Be colorful
- Extend the audience by making the portlet self-service
- Get user feedback and keep improving the portlet

- Get executive buy-in
- Eliminate duplicate forms of information distribution
- Think about the entire process, especially where to store and how to update data
- Do not try to make everyone happy
- Utilize portlet and web service activity rights: If you want users to be able to create portlets only from frameworks, you should give them only the Create Portlets activity right. For example, if you are using the Excel Portlet Framework, the users could create hundreds of portlets from a single Excel portlet web service, each differentiated by the administrative preferences.

If you want to allow users to create completely custom portlets, give users Create Web Service Infrastructure activity right, allowing them to create remote servers and portlet web services.

- Do not rely solely on activity rights: Users do not need rights to create portlets in order to edit portlets. You should make sure that the ACLs on the portlets and all the portlet components are correct.

Also, community owners have rights to create portlets from templates even without the Create Portlets activity right, so make sure that the templates are secured if you do not want community owners to create portlets.

- Secure templates and web services: Control access to templates and web services to restrict what type of portlets users can make. If users only have rights to make portlets from templates and web services, they can only make portlets from the templates and web services to which they have access.
- Use portlet bundles: User portlet bundles to simplify adding groups of related portlets to pages. A portlet bundle simply adds portlets already existing in the system.



## Creating a Rich Enterprise Web Experience

The goal of this section is to:

- Define Enterprise Web applications
- Summarize non-portlet portal user interfaces such as Search and Knowledge Directory (presentation modes)
- Summarize how to effectively exploit packaged Enterprise Web Suite features to create a rich Enterprise Web experience (sourcing modes)
- Give examples of how to go beyond packaged features and use the Enterprise Web Development Kit (EDK) to harness existing applications, packaged and custom, and provide effective solutions to typical business problems

### What Are Enterprise Web Applications?

Portlets have traditionally been viewed as convenient windows into back-end systems. In other words, they have been instrumental in increasing the reach of back-end systems to all relevant users rather than just a few select power users of these systems (for example, an order status portlet from the ERP system for use by all customers).

Alternatively, they have been considered vehicles to complete one simple task or deliver a specific piece of information (for example, a company approved holidays portlet for use by all employees).

You can build a lot of effective functionality and solve many business problems using the Enterprise Web's packaged functionality, such as building a simple inventory tracking system using Studio Server or a customer survey using Content Server. However, you also have the flexibility to integrate existing applications and build entirely new and composite applications.

By using the Enterprise Web and the Enterprise Web Development Kit (EDK), you can create applications that can rely on back-end systems such as ERP and CRM systems for data and document management systems for documents, but you can also include custom navigation menus, multiple application screens or pages, and even distinctive branding. These applications are called Enterprise Web applications.

An Enterprise Web application has the following characteristics:

- **Application:** It is a software application—a coherent collection of software features that solves a well-defined business problem. It might have its own custom navigation as well as distinctive branding.
- **Delivered via the Enterprise Web:** It is delivered to the end-user through the Enterprise Web, though it may be branded distinctly.
- **Composite:** It is often composite in nature, that is, its features are derived from the features of multiple back-end systems as well as Enterprise Web foundation services such as collaboration and content management.
- **User driven:** It provides users with the shortest, most convenient way to perform tasks that involve multiple systems and provides an intuitive Web user interface. In doing so, it buffers the average enterprise user from having to master multiple, complex back-end systems and access them individually. It is also often driven by a specific set of related user needs that have not yet been addressed.
- **Responsive:** It is meant to be built rapidly and if business needs so demand, be changed rapidly as well.

What an Enterprise Web application is not:

- By itself, it usually does not constitute a system of record.
- It is not an information island unto itself.
- It is not meant as a wholesale replacement for the power user interfaces of traditional back-end systems.

### *Portlets and Enterprise Web Applications*

The portlet is still the basic visual and operational unit of any Enterprise Web application. A typical Enterprise Web application consists of a number of portlets spread across the pages of one or more communities. You can achieve a distinctive branding or look by making the application part of a subportal. An Enterprise Web application can use the standard navigation bar consisting of communities and pages or use a custom pluggable navigation bar. Irrespective of whether an Enterprise Web application appears to be contiguous with the Enterprise Web or appears to be standalone, it has access to and is driven by all the useful

constructs that are part of the Enterprise Web—users, authentication, documents, secure access, portlets, and so on, and foundation services such as search, content management, and collaboration.

There is no magical transition point when a collection of portlets in a community or a set of communities turns into an Enterprise Web application. Rather, it is a continuum where one end consists of a collection of portlets, that might, at best, be thematically connected and the other end where the application has all the characteristics detailed above.

## Portal User Interfaces

Here is a summary of ways to present information using the Plumtree Corporate Portal:

- **Portlets:** This is perhaps the most common way of accessing information. Portlets can be present on My Pages or communities. Information conveyed by portlets can be derived entirely from a remote system or from information within portal objects or a combination of the two. The finished contents of a portlet can be cached by the portal but are nevertheless not stored in the portal for later use.
- **Gatewayed pages:** This is a way to tap into existing Web applications without having to sign on again. Typically, clicking on a link inside a portlet results in a pop-up window that surfaces a specific Web page from a Web application. The user is logged on behind the scenes by the portal, which passes the user credentials to the Web application. The portal then delivers the HTML page provided by the Web application to the pop-up window. There are a few restrictions on the types of Web pages that can be gatewayed in this manner, but, by and large, Web application pages can be gatewayed. The contents of a gatewayed page are transient, that is, not stored in the portal.
- **Portal search:** Simple search is available by using the keyword text box in the portal banner. Advanced search is available through an icon in the portal banner; in addition to searching for keywords, advanced search allows you to specify the type of object for which you are searching (for example, community or Collaboration Server project). Search results are listed in order of relevance. Each result corresponds to a document in the Knowledge Directory or an object in Collaboration Server or Content Server.
- **Federated search:** Federated search is also available through an icon in the portal banner. Like any other search, it takes in keywords and provides results. However, feder-

ated searches are carried out on remote systems and the results are conveyed back to the portal to display. The portal does not store these results.

- **Knowledge Directory:** The Knowledge Directory is accessed through the portal menu. It provides a hierarchically organized set of folders and links to documents. Knowledge Directory links can refer to documents that are uploaded into the portal or documents in a remote system. Each document has a set of properties that can be viewed.
- **User Profile/My Account:** Users can access profile information by clicking **My Account** in the portal banner.
- **Help pages:** Context-sensitive and conceptual online help is available through an icon in the portal banner (or in portlet title bars).

## Sourcing from Common Sources

In the previous section, we listed the various ways in which information gets presented through the portal. Table 2-3 catalogs the various ways of sourcing all the information. Sourcing and presentation modes are closely coupled.

In this table are examples of integration web services such as authentication, profile, search, and crawler web services. These web services are different ways of sourcing information and features from existing enterprise applications.

*Table 2-3: How Web Services Source and Present Information (Sheet 1 of 3)*

Information/ Feature Source	Description	Sourced Through	Presented Through
Plumtree Collaboration Server	Provides collaboration projects that can be used by groups for ad-hoc collaboration	Pre-packaged portlets and pages for project, document, discussion, calendar, and task list	Portlets, gate-wayed pages, portal search
Plumtree Content Server	Provides the ability to create and publish Web content using templates	Pre-packaged portlet templates, for example, forms and portlets for content workflow (editing, approval, and so on)	Portlets, gate-wayed pages, portal search
Studio Server	Provides the ability to create simple data-driven applications which require only one table	Pre-packaged portlets for creating Studio Server applications, viewing and searching for information, and submitting new data rows	Portlets, gate-wayed pages

*Table 2-3: How Web Services Source and Present Information (Sheet 2 of 3)*

<b>Information/ Feature Source</b>	<b>Description</b>	<b>Sourced Through</b>	<b>Presented Through</b>
Groupware integration products	Provide the ability for users to access e-mail, contacts and calendar-ing functions from Microsoft Exchange and Lotus Notes	Pre-packaged portlets	Portlets
Crawler integration products	Provide the ability to catalog and index external sources of data and documents such as NT files systems, Microsoft Exchange, and Docu-mentum	Pre-packaged crawlers using Plumtree crawler web services	Knowledge Directory, por-tal search
ERP, CRM integration products	Provide the ability to create portlets from systems of record such as SAP and People-Soft, by tapping into the business compo-nents of these systems	Pre-packaged portlets for commonly used features, configurable portlets for other business components	Portlets

*Table 2-3: How Web Services Source and Present Information (Sheet 3 of 3)*

<b>Information/ Feature Source</b>	<b>Description</b>	<b>Sourced Through</b>	<b>Presented Through</b>
JSR-168 integration portlets	JSR-168 is a standard for portlets. Vendors such as Documentum and Stellent have or are building JSR-168 compliant portlets which surface key document features such as check-in, check-out, and simple workflow.	Pre-packaged portlets	Portlets
Excel integration product	A framework for surfacing tables and graphs from Excel files within portlets	Configurable portlets	Portlets
LDAP and Microsoft Active Directory	Integration products that enable the importing and synchronization of users and their profiles, as well as groups from identity management systems such as Microsoft Active Directory and LDAP	Product is built on Plumtree authentication and profile web services	Login page, User Profile/My Account page
Microsoft Office	Note that the NT File crawler indexes all commonly found documents, including Microsoft Word, Powerpoint, and so on		

## Sourcing from Custom Sources

Providing pre-packaged integration for all but the most ubiquitous packaged software is not a scalable proposition. Also, it is clearly not possible for Plumtree to provide pre-packaged integration for custom applications. However, the need to tap into these applications to deliver information and features to portal users is very real.

This is where the Enterprise Web Development Kit comes in. It exists to enable developers to execute on well-defined business use cases for integrating into existing applications, packaged or custom, as well as build new applications to meet the needs of a dynamic enterprise.

We offer here a brief glimpse of the functionality inherent in the EDK by describing business scenarios where the functionality is successfully used. For more information, refer to the *Enterprise Web Development Guide* or online resources available in the Development Center at <http://portal.plumtree.com>.

### Example 1: Access and Personalization

Suppose your customer accounts are set up in a CRM system such as Siebel. As new accounts are added to Siebel, you need to provide customers access to the portal and certain communities on the portal.

You can set up an authentication web service to enable the portal to periodically connect to the Siebel system, query for customer users, and provision accounts for them on the portal. In addition, this authentication web service can also be configured to accept the username and password at login and authenticate the user against Siebel. This eliminates the need for creating portal accounts manually. Also, since access is based on the back-end system, the moment the customer account is suspended on Siebel, access on the portal is also automatically denied.

You can also set up a profile web service in conjunction with the authentication web service to tap into each customer's profile information on Siebel and augment the portal user profile. This profile information can be used to add the customer user to the appropriate groups that will drive personalization. This profile information might be the basis for providing access to relevant communities; for example, a company that sells wines could pro-



vide their distributor with access to the appropriate community depending on their geographical address, a portlet that lists all the special events could be personalized based on the user profile information that tracks geography.

## Example 2: Searching for Data and Documents

Suppose that you want to provide a way for customers to look at the status of their orders and these orders are stored in a Lawson ERP system (it could be in any ERP system, packaged or homegrown). One solution is to develop a custom search web service that, given keywords such as start and end dates, pulls in a set of sales orders and associated status information. The search web service would use the customer user credentials and information for identification purposes and return only those sales orders corresponding to that customer. This search could be conducted from the Federated Search page or, alternatively, set up as a portlet inside a community.

Suppose that you want to be able to provide the customer with access to all the contracts that they have signed with your company and that these contracts are stored in an NT file server. You can use the Plumtree Crawler Web Service for NT File Systems to crawl these documents into the Knowledge Directory, and the customer can browse the Knowledge Directory to find the contracts. Alternatively, you can create a content snapshot portlet that provides them with links to the Knowledge Directory that correspond to the contracts that pertain to them.

What if instead of just finished contracts, you need to keep various versions of contracts that are being negotiated between your sales representative and the customer. Use a Collaboration Server project to store these documents. Collaboration Server projects are appropriate for work in progress. When the document is finished, it can be stored in a document repository.

## Example 3: Customer Branded Support Site

You can choose to give each customer a personalized experience replete with custom branding. You can create a subportal that includes the customer's own branding. As soon as the customer logs in to the portal, they are directed to their branded subportal. You can build custom navigation using the EDK's Pluggable Navigation, replacing the portal's regular

navigation. The look of a subportal can be completely different from that of the parent portal.

Suppose that whenever a user from a particular customer site logs on, you want to be notified. Using the EDK's Portal Event Interface (PEI), you can write custom code that gets triggered upon the customer logging in and notifies you via e-mail.

# 3

## Technology Foundation

This chapter is written for IT personnel tasked with installing and configuring the Plumtree Enterprise Web Suite. It describes how to align your network and hardware to the Enterprise Web, making sure that it performs reliably, scales as needed, and provides adequate security for your business.

The power, versatility, and flexibility of the Plumtree Enterprise Web Suite make it difficult to state general rules regarding site planning. However, you can create a quality plan by collecting as much information as possible about your expected usage patterns, availability, performance, and administration requirements.

### Configuration Options

The Plumtree Enterprise Web Suite consists of a variety of Web applications and back-end services that work together to provide portal, collaboration, content publishing, and search functionality. Planning a configuration, whether it is a small proof of concept or a large highly-available production system, requires both an understanding of factors for which the system is being optimized and the functions and resource requirements of all components being deployed. This section describes the factors for which you might optimize a deployment and a number of possible configuration options based on the number of available host machines. It complements the hardware sizing and scalability sections later in this chapter, which go into more detail about estimating load and capacity to define the scope of a Plumtree project.

Since there are many components to a Plumtree Enterprise Web Suite deployment, and many factors that influence how you configure the system, let us begin by listing a set of goals for which you might optimize a deployment:

- **Low initial hardware cost:** Companies optimizing for low initial hardware cost seek to buy the least expensive machines necessary to make the software work reliably. Given a choice between repurposing two existing 1x700 MHz Pentium III servers and spending \$7,500 on one 2x2.4GHz Pentium IV Server, they would choose the former.

- **Low hardware maintenance cost:** Companies optimizing for low hardware maintenance costs seek to reduce the number of machines needed to host the software. Because each additional computer incurs a minimum fixed cost in terms of administrative overhead, power consumption, space, and operating system license, these companies would rather combine multiple Plumtree components on a single, more powerful computer than distribute those components over multiple, less expensive machines.
- **High availability:** Companies optimizing for high availability are willing to spend extra money and effort to ensure that the portal and other Plumtree components are available reliably to their users at all times. Such companies typically purchase more computers and load balance them where possible, creating redundant configurations.
- **Low software maintenance cost:** Companies optimizing for low software maintenance cost assume that at some point in the life of the system, some part of the software will malfunction, and they seek both to lessen the chance that malfunctions will occur and lessen their impact when they do occur. Such companies would typically purchase more individual computers to ensure that system components do not interfere with one another, and to reduce the risk that taking a computer out of the system to install new software will impact multiple system functions.
- **Scalability:** Companies optimizing for scalability assume that their deployments will be required to handle a large number of users. Such companies would typically purchase extra hardware, and more expensive hardware, in order to create excess capacity in the system.
- **Performance:** Companies optimizing for performance seek to make their systems operate as fast as possible, especially in their ability to render pages quickly for end-users. Like companies seeking to lower software maintenance costs, these companies would distribute system components across a larger number of computers to ensure that each component has unrestricted access to the computing power it needs to perform its tasks the moment those tasks are called for.
- **Network Security:** Companies optimizing for network security seek to ensure that end-users touch only machines hosting the smallest amount of code and data. Such companies also typically install firewalls between layers of their deployment, to ensure that if an intruder compromises one layer, the potential damage is limited. Such companies tend to purchase more computers in order to isolate the Portal Server, which end-users touch directly, from other components.

Of course, everyone wants a deployment that is highly available, perfectly reliable, fast, secure, cheap to maintain, and runs on the minimum amount of hardware. Fortunately, many of these goals go hand-in-hand; creating a system with lower software maintenance costs lends itself well to creating a system that is secure and performs well. Unfortunately, most companies do not have unlimited money; almost everyone seeks to minimize hardware costs, both initial purchase costs and ongoing maintenance costs. Planning a deployment therefore involves prioritizing your requirements and sometimes involves making trade-offs. Minimizing the number of computers used in the deployment will, for example, tend to make the system harder to debug and less performant. We will discuss ways to mitigate those risks when combining components on a single machine.

In addition to these optimization factors, deployments differ in the system components they emphasize: some deployments have a large number of users, but not very much Knowledge Directory content; other deployments have hundreds of thousands of documents, but a relatively small number of users; other deployments rely upon CPU-intensive portlets such as e-mail portlets; still others focus on collaboration. A deployment's emphasis will impact how resources are allocated.

Keeping their goals and costs in mind, companies beginning the deployment planning process should consider these general rules of thumb:

- User capacity is not an issue for most companies. Companies who have 5,000-10,000 users, most of whom use the system for five or six sessions a day will probably not have a problem with capacity if they use modern hardware in a four or five machine configuration. Even companies with tens of thousands of users do not typically need to worry about capacity planning if the users hit the system only infrequently. Capacity planning is more of an issue for very large companies, for companies who expect extremely heavy usage, and for companies who import hundreds of thousands of documents into the Knowledge Directory and refresh them often. Most companies do not do this, and should therefore optimize their deployments for other goals, such as reliability or lower hardware cost.
- The most important factor in capacity planning is how much users hit the system, not how many users there are. If you have 1,000 users hitting the system 20 times per hour, you will require much more capacity than 20,000 users hitting it once a day. The number of hits the system gets is also far more important to capacity planning than the

number of concurrently active sessions. For example, 1,000 users hitting the system 60 times per hour (60,000 hits/hour) will require much more capacity than 5,000 users maintaining an active session, but only hitting the system once every twenty minutes (15,000 hits/hour).

- The requirements for a development system might be vastly different from those of a production system. Co-locating multiple components on a single computer might provide acceptable reliability and capacity for your development or staging system. Requirements for a production system are typically more stringent, so more computers might be necessary.
- More computers, rather than larger computers, tend to produce a more reliable, available, and performant system. Computers that serve one purpose in a deployment tend to experience fewer problems, and tend to be easier to debug when problems occur, than computers that serve multiple purposes. Computers with excess capacity tend to perform better than computers running at the upper end of their load curves. When planning their systems, many companies ask, “Do I really need five computers to serve 1,000 users?” The answer is typically, “No, but you may want five computers to ensure a high level of reliability and performance, and to make it easier to troubleshoot and maintain the system.”
- For the Portal Server, two 2-CPU computers provide more capacity than one 4-CPU computer. A 4-CPU machine has its place but is more valuable hosting the database or the Search Server than the Portal Servers, Collaboration Servers, or most Remote Servers.

## Components Involved in an Enterprise Web Suite Deployment

Now that we understand the factors for which you might optimize a deployment and some general rules of thumb, let us discuss the different components involved in an Enterprise Web Suite deployment and the factors that impact their reliability, performance, scalability, and network security. We will discuss specific sizing and scaling options in a later section of this chapter.

### Web Applications

The following components are Web applications deployed within an Application Server:

- **Portal Server:** The Portal Server provides the Web interface through which users interact with all components of the Enterprise Web Suite. The portal's dynamic pages are generated through compiled C# or Java classes.

*Reliability:* The Portal Server computer must be up and running smoothly at all times because users hit it directly. To ensure predictable, reliable Portal Server performance, Plumtree recommends that for most deployments, the Portal Server component be hosted on computers separate from all other components. This is the case because other components touch back-end systems whose reliability may be highly variable. Load balancing multiple Portal Servers further increases reliability. If one computer goes down, requests fail over to the other machines in the Web farm. Additionally, load balancing allows you to install software patches and upgrades without bringing down the system, since one computer at a time is taken out of the rotation.

*Performance:* Isolating the Portal Server from other components will improve performance. Combining the Portal Server, which responds synchronously to user requests, with components such as the Automation Server and the database, whose load increases during the processing of asynchronous tasks, tends to degrade response times for end-users when synchronous and asynchronous tasks occur at the same time. More RAM on a Portal Server increases performance because the server caches a large amount of data, but there are few benefits to having more than 2 GB of RAM on a Portal Server.

*Scalability:* Portal Servers scale horizontally—that is, you increase capacity by adding more computers—and scale linearly as new computers are added to a Web farm. More CPUs on a Portal Server give it the ability to handle more hits, and therefore more users. While we will discuss Portal Server capacity in detail later, most companies' needs are accommodated by two modern 2-CPU Portal Servers, load balanced in a Web farm.

*Security:* Separating the Portal Server from other system components increases security since persistent data (search and database) and back-end tasks (Automation Server) are not on the same computer. Additionally, companies running .NET portals in extranet environments typically host the Portal Server on its own computer and place that computer in a DMZ, with all other components (except the Image Server) hosted behind the internal firewall.

- **Administrative Portal Server:** The Administrative Portal Server is simply a Portal Server that also includes the administrative Web application. Many, or even most, companies deploy only Administrative Portal Servers rather than separating administration from end-user access. There are two reasons to separate administration from end-user usage:

*Performance:* Some administrative actions are very CPU-intensive and could degrade end-user performance. Companies can mitigate this risk by using multi-CPU machines. Note, also, that Administrative Portal Servers serve very few users even in large deployments. These users are also, by definition, savvier and probably more tolerant of slow performance. Administrative Portal Servers that are not also acting as general Portal Servers may therefore be combined with Automation Servers or other components with little risk.

*Security:* Some companies prefer that end-user facing machines not have any administrative capabilities, and that administrative machines be physically separated from potential intruders.

- **WS Server:** The WS Server provides access to the Plumtree SOAP API. Separating the the SOAP API to its own server allows for greater flexibility in deployments, especially for extranets. However, most companies host the WS Server on the same computer as the Portal Server.



*Reliability:* Unless you make extensive use of the SOAP API and program sloppily, the WS Server should not greatly impact the overall reliability of the system and can be combined with other components.

*Scalability:* Unless you make extensive use of the SOAP API, the scalability of the WS Server should not bottleneck the system.

*Performance:* SOAP transactions tend to be slower than many other kinds of computer-to-computer communications. However, unless you develop Enterprise Web Applications that make extensive use (on the order of thousands of transactions per hour) of the SOAP API, you should not encounter performance issues in the WS Server.

Plumtree's server products use the SOAP API and their use will not typically create performance issues with the WS Server.

*Security:* Access to the SOAP API requires a valid Plumtree session, which is instantiated through a token that expires after a configurable period of time. It is therefore difficult to gain malicious access to the system through the SOAP API. However, your company might have policies against exposing a SOAP API through the extranet, in which case you can separate the WS Server from the Portal Server, placing the Portal Server in the DMZ and the WS Server behind the inner firewall.

- **Collaboration Server:** Collaboration Server provides project collaboration, including document sharing, calendars, and threaded discussions. Collaboration Server uses J2EE technology and connects to the Portal Server through portlets via HTTP, to the WS Server via HTTP/SOAP, to its own and the portal's database via JDBC, and to the search server via TCP.

*Reliability:* In general, the overall reliability of the system will be higher if Collaboration Server is separate from other components. Separation of components precludes any contention within the JVM process on the computer. To improve reliability, Collaboration Server can be load balanced. When optimizing for a smaller number of machines, you can combine Collaboration Server on the same Remote Server (discussed later) with other components that generate portlets, including Studio Server and Content Server.

*Scalability:* Collaboration Server scales horizontally, so more Collaboration Servers can be added to increase capacity when necessary. Most companies will be well served by a modern 2-CPU server running Collaboration Server.

*Performance:* Collaboration Server should not have a great impact on the overall performance of the system. For best performance, the computer hosting Collaboration Server should have at least 1 GB of RAM.

*Security:* There are no special security issues associated with Collaboration Server. Since end-users do not touch the system directly (all access to it is gatewayed through the Portal Server), it can reside behind the inner firewall of a DMZ.

- **Content Server:** Content Server provides Web publishing capabilities, including versioning, content workflow, and Web content management. Like Collaboration Server, Content Server uses J2EE technology and connects to the Portal Server through portlets via HTTP, to the WS Server via HTTP/SOAP, to its own and the portal's database via JDBC, and to the Search Server via TCP. Content Server publishes HTML pages to a Web Server and can also transfer images to a Web Server via file copy or FTP. This Web Server, known as a “publishing target,” can be co-located with Content Server or hosted on a separate machine.

*Reliability:* In general, the overall reliability of the system will be higher if Content Server is separate from other components. Separation of components precludes any contention within the JVM process on the computer. When optimizing for a smaller number of machines, you can combine Content Server on the same Remote Server (discussed later) with other components that generate portlets, including Studio Server and Collaboration Server.

*Scalability:* In a typical deployment, the number of users who contribute and approve content is a small fraction of the total number of registered portal users. Also, once the content is published, it is static, so that a large number of users can access HTML pages, files, and images published through Content Server without much incurred processing overhead by the Content Server Web application itself. Content Server scales vertically in version 5.0.2. If Content Server capacity is an issue, place it on its own computer, and purchase more powerful hardware. Most companies will be well served by a modern 2-CPU server running Content Server.

*Performance:* Under most circumstances, Content Server should not have a great impact on the overall performance of the system. For best performance, the computer hosting Content Server should have at least 1 GB of RAM, since Content Server caches a great deal of data in memory. Performance might suffer somewhat if Content Server shares a computer with other components, since it will not have as much RAM to work with. Certain Content Server operations, particularly publishing, make heavy use of the back-end database. If you are publishing large volumes of information through Content Server, you should monitor the capacity of the computer hosting your database.

*Security:* There are no special security issues associated with Content Server itself. Since end-users do not touch the system directly (all access to it is gatewayed through the Portal Server), it can reside behind the inner firewall of a DMZ. Optionally, the HTML pages published by the system can be made accessible to end-users directly, rather than through the portal gateway.



**Note:** Plumtree Corporate Portal 5.0 and 5.0.1 include a subset of the Content Server code, called the Branding Engine, that provides templates for communities and portlets; it is used in deployments that do not include Content Server.

- **Studio Server:** Studio Server provides the ability to build small applications using portlet technology without the use of developed code. You can build portlets using the Studio Server framework and then make them available to users through their My Pages and communities. Like Collaboration Server, Studio Server uses J2EE technology and connects to the Portal Server through portlets via HTTP, to the WS Server via HTTP/SOAP, and to its own and the portal's database via JDBC.

*Reliability:* In general, the overall reliability of the system will be higher if Studio Server is separate from other components. Separation of components precludes any contention within the JVM process on the computer. When optimizing for a smaller number of machines, you can combine Studio Server on the same Remote Server (described later) with other components that generate portlets, including Collaboration Server and Content Server.

*Scalability:* Studio Server scales vertically, so companies can add more capacity by separating Studio Server from other components and by locating it on a more powerful computer. Most companies will be amply served by a modern 2-CPU server running Studio Server.

*Performance:* Studio Server should not have a great impact on the overall performance of the system. Studio Server makes extensive use of in-memory caching. For best performance, the computer hosting Studio Server should have at least 1 GB of RAM. Performance may suffer somewhat if Studio Server shares a computer with other components, since it will not have as much RAM to work with.

*Security:* There are no special security issues associated with Studio Server. Since end-users do not touch the system directly (all access to it is gatewayed through the Portal Server), it can reside behind the inner firewall of a DMZ.

You can deploy all of the above components in a Java Application Server (Tomcat, BEA WebLogic, or IBM WebSphere). You can also deploy the Portal Server, Administrative Portal Server, and WS Server in IIS using .NET.

## Static Web Components

- **Image Server:** The Image Server stores static content (such as images, JavaScript, and online help files) for use by the Enterprise Web Suite. End-users touch the Image Server directly to receive images and static content. Most companies simply host the Image Server on the same computer and in the same application server that hosts the Portal Server. If you are serving a high volume of users, you can host images and other static files on a separate machine or in a separate Web server on the same machine. Separating the images from the Portal Server will improve Portal Server throughput by approximately 10%. If you are seeking to conserve bandwidth, you can consider using a separate Image Server to locate images closer to end-users.

## Services

The following components are back-end services that do not require an Application Server:

- **Automation Server:** The Automation Server provides asynchronous services to the portal, such as crawling, index updating, and user account and group synchronization. It can also perform custom scheduled operations. In Windows deployments, the Automation Server runs on Windows 2000 as a Windows service. In Unix deployments, the Automation Server runs as a daemon or console process. It connects to other components through ODBC or through other TCP protocols.

*Reliability:* The reliability of the system is significantly enhanced when the Automation Server is separated from other components. Because many of the Automation Server's tasks involve connection to external systems whose reliability may be questionable, the Automation Server might have different performance characteristics at different times. This variability suggests separating the Automation Server from components that serve end-users. Companies can load balance Automation Servers using a round-robin technique, registering a single job to run on multiple Automation Servers. If one Automation Server goes down, a second will pick up the job.

*Scalability:* You can use multiple Automation Servers if you have a large number of intensive jobs. Individual Automation Servers can handle more concurrent jobs with more memory and perform jobs faster with more and faster CPUs. Plumtree recommends two CPUs and 1 GB of memory for an Automation Server computer for medium or larger deployments, with more CPUs and more memory if you plan to crawl a large number of documents into the Knowledge Directory.

*Performance:* For production deployments, Plumtree strongly recommends separating the Automation Server from other components, particularly the Portal Server. Automation Servers are responsible for a great deal of processing that imposes a burden on the Automation Server's CPUs. Operations such as user and group imports and synchronizations and content crawls and refreshes are particularly taxing. While these operations are being performed, the performance of other activities running on the same computer might degrade. End-users relying on that computer to render pages or host Collaboration Server projects would therefore experience decreased and potentially unacceptable performance. If you are seeking to minimize the number of computers in your

deployment and are considering combining the Portal Server with the Automation Server, you can mitigate the risks by scheduling all of the Automation Server's tasks to run in off hours. This is particularly practical if you are not importing large volumes of content into the portal's Knowledge Directory.

*Security:* There are no specific network security issues with respect to Automation Servers. Most companies locate the Automation Server inside the inner firewall of a DMZ, and many restrict access to the Automation Server to specific computers.

- **Document Repository:** The Document Repository is used by the Portal Server (starting with version 5.0.2, for document upload), Collaboration Server, and Content Server to store their content. In Windows deployments, the Document Repository runs on Windows 2000 as a Windows service. In Unix deployments, the Document Repository runs as a daemon or console process. Since the primary function of the Document Repository is storage, it should have ample space on a reliable disk, otherwise there are no particular performance or reliability issues associated with it. For security purposes, companies should keep the Document Repository behind the inner firewall of a DMZ, and should consider restricting access to the computer to only other portal components. End-users never access the Document Repository directly.
- **Search Server:** The Search Server provides indexing and query services for Enterprise Web content and portal objects. It runs on Windows 2000 as a Windows service or on Solaris as a daemon. It connects to other components through a TCP network protocol.

*Reliability:* The Search Server stores and manages data and should ideally be treated like a database and hosted on its own computer. In situations where Search Server reliability is particularly important, you can improve reliability in two ways: splitting the Search Server into an indexing server and a query server; and instituting fail-over between two replicated Search Servers. Having unrestricted access to its own files is particularly important to Search Server reliability, so you should take care to ensure that third-party products, including virus detectors, do not attempt to access Search Server files while the system is running.

*Scalability:* Available RAM is the most important factor in how much text the Search Server can index, while CPU requirements determine how many users it can serve. For very large collections (for example, those with more than 500,000 documents or more than 6.25 GB of raw text) a 64-bit Solaris Search Server should be considered. Solaris offers the best stability and scalability in terms of RAM expansion. For smaller collections, however, the cost of a Solaris server cannot be justified, and a Windows server should be sufficient.

CPU requirements for the Search Server are largely a function of the number of portal users. Generally speaking, the Search Server is highly scalable in terms of query speed. Plumtree has performed scale experiments with a test collection of 110,000 documents averaging 28 KB, for a total of 3.1 GB of raw text indexed. Our experiments indicate that a 4-CPU@700Mhz system is capable of handling up to 156,000 portal users, under reasonable assumptions about the frequency of usage of search (these assumptions are described in [“General Assumptions About the Search Server Environment” on page 3-49](#)). A 2-CPU@700Mhz system would be sufficient for up to 78,000 users.

*Performance:* Indexing and querying are memory intensive, and the Search Server makes extensive use of memory mapping for efficient file I/O. As a result, it is extremely important to have at least 3 GB of paging file or swap space to support virtual memory operations. In addition, it is important to have an adequate amount of physical RAM to support internal caching done by the Search Server. (Internal Search Server caching is completely separate from file system or portal/portlet caching).

Because file I/O is such an important part of indexing and querying operations, the Search Server needs high-end disk hardware. Installing the Search Server on low-latency disk drives or disk arrays is important for performance. Any kind of fast, local disk is acceptable (for example, SCSI, RAID arrays), and there are no known restrictions with respect to hardware or OS disk caching.

*Security:* Since the Search Server stores data, most companies place it behind the inner firewall of a DMZ and restrict the computers that can access it to those hosting other Plumtree components.

## Database Services

The following components are SQL Server 2000 or Oracle 9i persistent storage used by the Web applications and Automation Server:

- **Portal Database**
- **Collaboration Server Database**
- **Content Server Database**
- **Workflow Database** (only used if you install Content Server)
- **Studio Server Database**
- **Analytics Server Database**

Plumtree has observed significant performance differences between Oracle and SQL Server, with Oracle being slower at almost all operations. Given equal hardware for the database, companies using Oracle will achieve 15-20% fewer hits per second on the Portal Server, and write-intensive user synchronization, crawling, and logging will also be significantly slower.

## Remote Servers

Remote Servers are any computers hosting remote web services such as portlet, crawler, and authentication web services, as well as the Plumtree Analytics Server. There are many different types of web services, and deployment options for Remote Servers hosting those web services depend on the specifics of the web service being hosted. In general, the system was designed so that those services would be hosted remotely from the Portal Server to improve reliability and flexibility.

You can host Remote Servers on any HTTP serving platform. Plumtree provides the Enterprise Development Kit to assist with web service development in ASP.NET and Java.

Remote Servers are only loosely coupled to the portal system. Outside groups can manage and maintain their own Remote Servers with only a minimal registration in the portal, and other responsibilities can be delegated to the outside groups.

For the Plumtree Analytics Server, you must enable Unicast UDP on port 31314 for communication between the Analytics Server agents installed on the Portal Servers and the Analytics Server Remote Server.



**Reliability:** Reliability is the most important reason not to co-locate web services with the Portal Server. Many remote services require the installation of additional, third-party software on the computer. This third-party software can interfere with the portal software in unpredictable ways. Companies might write portlets or other services and introduce errors. Keeping this code separate from the Portal Server improves reliability and makes it much easier to debug issues when they arise.

Because of conflicting versions of DLLs, some crawler web services cannot share a computer with the Portal Server, Automation Server, or specific other web services. Refer to the documentation and release notes that come with the crawler web service for specific limitations.

**Scalability:** Different types of web services and different web services of the same type have different scalability implications. Portlets can typically be scaled using the parallel portal engine's native load balancing capabilities or using application server load balancing if the portlets support this. The scalability of individual portlets varies widely. In general, static portlets with minimal personalization scale very well to any number of users; dynamic portlets with more personalization require more processing power on the Remote Server because they cannot be cached as effectively. Asynchronously operating web services, such as crawler web services and authentication web services, impose more of a scalability burden on the Automation Server than on the Remote Server.

**Performance:** The system will perform best when portlets are separate from the Portal Server. Much of the performance of portlets depends on the back end system that they connect to, the speed of the network connection between the back-end server and the Remote Server, and whether caching is invoked on the Portal Server.

**Security:** Flexibility in network security is one of the foremost advantages of an architecture based on web services. Since web services connect to the Portal Server via HTTP(S), Remote Servers can be physically separate from the Portal Server: behind a firewall, in a different domain, or even across the Internet. It is possible, therefore, to host a crawler web service in the London office, host the Automation Server in New York, and connect the two over the Internet. In general, for asynchronous web services such as crawler and authentication web services, it is preferable for performance reasons to host the web service as close to the source repository as possible.

## Consolidating Components

If you are running production systems optimized for maximum reliability, performance, scalability, ease-of-troubleshooting, and availability you will tend toward more (but less powerful) computers. Since cost of hardware, and maintenance of hardware, is frequently a factor, many companies cannot use as many computers as Plumtree might recommend. Consolidating components thus becomes desirable.

Some components can be consolidated with little risk, and the risks of consolidation can sometimes be mitigated. The following section describes options for deploying the Enterprise Web Suite with one to five computers, thereby consolidating multiple components onto single computers. These examples do not take into account the number of users, nor do they include every portlet available from Plumtree or third parties. They are meant to provide guidelines, and to show the impact of choices you might make in you deployment.

First, here are some general rules of thumb for consolidating components:

- Most companies do not need to separate Portal Servers, Administrative Portal Servers, and Image Servers. Separating these components is most useful for larger deployments or extranet deployments where security policies dictate that administration not be accessible to the extranet. In most cases, reliability should take precedence, and you should combine those components and deploy a load-balanced configuration. Using multi-CPU computers further reduces any risks involved.
- Consolidate web services on a single computer when possible. Most portlets will perform reliably when they share a computer with other portlets, and most can share a computer with other types of web services. When trying to minimize the amount of hardware employed, Content Server, Collaboration Server, and Studio Server can share a computer with one another and with portlets. If you are combining these products on a single computer pay special attention to the memory settings for the JVM. Refer to the Plumtree Knowledge Base for further details. Also, consider running the components on different ports to facilitate troubleshooting.



**Note:** E-mail portlets, particularly for large numbers of people, are CPU-intensive and may cause capacity problems when combined with other components.

- Combining the Search Server and the Database can make sense in some deployments. Although many DBAs will not allow other components to share a computer with the database, the Search Server and the database can successfully coexist on a large enough computer in a small to medium-sized deployment.
- Use SQL Server rather than Oracle. SQL Server 2000 provides better performance than Oracle 9i, both in the amount of CPU power required by the database and in the amount of CPU power required by the Portal Server. Crawling and user synchronization are significantly faster with SQL Server than with Oracle.
- Buy computers with plenty of RAM. When co-locating multiple components on a single computer, use at least 2 GB of RAM.
- Time can help you save money. Co-locating the Portal Server and the Automation Server is generally a bad idea. However, if your deployment does not have a large amount of content in the Knowledge Directory, does not synchronize a large number of users from an external user directory, and is largely inactive at night, you can schedule the Automation Server to do most of its work at night, leaving the Portal Server free to use system resources to serve users during the day. There are other cases where performing asynchronous tasks in off hours can help consolidate computers.

## One-Machine Configuration

You should never run all Plumtree components on a single machine for a production system. The Plumtree Enterprise Web Suite is fundamentally a distributed architecture, motivated by the desire for horizontal scalability and increased reliability. However, in circumstances such as a Proof of Concept, a development system, or small-scale demonstration, you might need to run all components on a single machine. In this case (and for other small-scale deployments consisting of 2 or 3 machines), you must be aware of the following points:

- Application server JVM settings

Consult your Java application server documentation for details on configuring the Java Virtual Machine for resource-constrained situations. The most important settings to be aware of are:

- `-Xms` - Initial heap size. You can reduce this value from the default value of 256 MB.
- `-XX:NewSize` - Size of the new generation memory pool for garbage collection. You can reduce this value from the default value of 64 MB.

- Search Server configuration file

The Search Server installation includes a set of example configuration files, which can be used to control process size in resource-constrained situations. The files are in **ptsearchserver\5.0\config**. There are demonstration, small, medium, large, and maximum examples, based on the amount of available physical RAM. To influence or minimize the memory profile of the Search Server, you can copy an example file and rename it `ignite.ini`—`ignite.ini` is the configuration file loaded by the Search Server. The example files are discussed further in [“Search Server Configuration File Sizing” on page 3-46](#).

- Databases

Consult your Oracle or SQL Server database documentation for details about configuring the database for resource-constrained situations.

## Omitting Components

Depending on the goals of the Proof of Concept, you can omit certain components from a single-machine install. At minimum, a portal deployment requires installation of the following components:

- Administrative Portal Server
- Image Server
- Search Server
- Portal Database

Then, additional components are required to add specific functionality:

- Automation Server - Required to import documents, users, groups, and user profile information into the portal.
- Content Server, Document Repository, Content Server Database - Required to provide template-based branding for portal communities and the full set of Content Server portlets and features.



**Note:** Beginning with version 5.0.2, you must include the Document Repository if you want to use document upload.

- Collaboration Server, Document Repository, Collaboration Server Database - Required for Collaboration Server portlets and features.
- Studio Server, Studio Server Database - Required for Studio Server portlets and features.
- WS Server - Required for access to the Plumtree SOAP API.
- Analytics Server, Analytics Server Database - Required for Analytics Server console and portlets.

## Two-Machine Configuration

As with the one-machine configuration, the distributed Plumtree architecture means that a two-machine configuration is suitable only for Proof of Concept demonstrations, and for development or testing systems. Because demonstrations typically involve a large amount of interactive work, in which rendering of My Pages and administrative screens is paramount, it is important to distribute the load for the Web Application components.

*Table 3-1: Two-Machine Configuration*

Machine	Components
1 (portal Web application and one back-end service component)	<ul style="list-style-type: none"><li>• Portal Server</li><li>• Image Server</li><li>• Automation Server</li></ul>
2 (additional Web applications and back-end service components)	<ul style="list-style-type: none"><li>• WS Server</li><li>• Collaboration Server</li><li>• Content Server</li><li>• Studio Server</li><li>• Document Repository</li><li>• Search Server</li><li>• Analytics Server</li><li>• Portal, Collaboration Server, Content Server, Workflow, Studio Server, and Analytics Server Databases</li></ul>

### Notes and Guidelines

- With such a minimal configuration, both machines should have significant memory and CPU resources. The two-machine layout presumes that both machines have similar technical specifications. If they have different specifications, all Web application components should be moved to the fastest machine with the largest number of CPUs.

- A machine hosting Web application components benefits from a lot of RAM and multiple high-speed CPUs. Disk space and speed are less important for this machine.
- A machine hosting back-end service components benefits from large amounts of RAM and high-speed disk. It also benefits from having multiple CPUs; however raw CPU speed is less important for this machine.

Risk and Risk Mitigation

A two-machine configuration could be suitable for user acceptance testing for 100-200 users, where system availability is not a priority. Assuming you wanted to deploy a two-machine system to a small number of users, the following table summarizes the risks and mitigating factors:

Table 3-2: Risks and Mitigating Factors

Risks	Suggestions
The Automation Server might draw CPU and memory from the Portal Server during crawls, maintenance operations, and user synchronization.	<ul style="list-style-type: none"><li>• Schedule the Automation Server to run all of its operations at night.</li></ul>
Collaboration Server, Content Server, Studio Server, Search Server, and Analytics Server and the database content for memory, slowing performance.	<ul style="list-style-type: none"><li>• Do not index a large number of documents, host many Studio Server databases, or publish a lot of content.</li><li>• Increase the amount of RAM on the second machine.</li><li>• Set all components to use a smaller memory footprint.</li><li>• If performance is still unacceptable, move the database to a separate server and follow the three-machine configuration option.</li></ul>

## Three-Machine Configuration

With three machines you reach the minimum configuration acceptable for a production portal system. If Collaboration Server, Content Server, or Studio Server components will also be installed, a three-machine configuration is still not advised for anything other than a Proof of Concept. What follows are two variations depending on whether the deployment emphasizes the number of users or the amount of content.

### Variation 1: Smaller Number of Users, Larger Amount of Content

*Table 3-3: Three-Machine Configuration, Variation 1*

Machine	Components
1 (all Web application components)	<ul style="list-style-type: none"><li>• Portal Server (with the Administrative Portal Server - unless otherwise noted “Portal Server” includes the Administrative Portal Server)</li><li>• Image Server</li><li>• WS Server</li><li>• Collaboration Server</li><li>• Content Server</li><li>• Studio Server</li></ul>
2 (search and indexing components)	<ul style="list-style-type: none"><li>• Automation Server</li><li>• Search Server</li><li>• Analytics Server</li></ul>
3 (persistent storage components)	<ul style="list-style-type: none"><li>• Document Repository</li><li>• Portal, Collaboration Serer, Content Server, Work-flow, Studio Server, and Analytics Server Databases</li></ul>



### *Notes and Guidelines*

- Because this variation emphasizes the amount of content (whether crawled into the Knowledge Directory, submitted through Collaboration Server, or published through Content Server), the back-end service components have been split onto separate machines.
- The machine hosting search and indexing components will benefit most by having a large amount of RAM, high-speed disk, and multiple CPUs. CPU speed is less important for the machine hosting the Databases. The machine hosting the Web applications will benefit from a large amount of RAM, followed by multiple CPUs.
- Use SQL Server rather than Oracle if possible. Performance will be better.



**Important:** For some three-machine deployments, and almost every deployment using an Oracle database, IT best practices might dictate that the database server be hosted, managed, and sized separately from all other portal components. In this case, you should use the two-machine configuration, using the third machine for the database and placing the Document Repository on the same machine as other back-end services like the Search and Automation Servers.

Variation 2: Larger Number of Users, Smaller Amount of Content

Table 3-4: Three-Machine Configuration, Variation 2

Machine	Components
1 (portal Web application components)	<ul style="list-style-type: none"><li>• Portal Server</li><li>• Image Server</li></ul>
2 (additional Web application components)	<ul style="list-style-type: none"><li>• WS Server</li><li>• Collaboration Server</li><li>• Content Server</li><li>• Studio Server</li><li>• Analytics Server</li></ul>
3 (back-end service components)	<ul style="list-style-type: none"><li>• Automation Server</li><li>• Search Server</li><li>• Document Repository</li><li>• Portal, Collaboration Server, Content Server, Workflow, Studio Server, and Analytics Server Databases</li></ul>

Notes and Guidelines

- Because this variation emphasizes the number of users interacting with the front-end Web application components, these components have been put onto separate machines.
- The machines hosting Web applications benefit most from additional RAM and multiple CPUs.
- Perform user synchronization and crawling at night. This will spare CPU cycles on the back-end machine during the day, improving performance for end-users browsing and searching the portal.

## Four-Machine Configuration

With four machines you reach the minimum configuration acceptable for a production Enterprise Web system consisting of the portal, Collaboration Server, Content Server, and Studio Server. At this stage you can separate the CPU- and memory-intensive Web applications from the memory- and IO-intensive services without unacceptable resource contention. However, this is still a minimal configuration and running the components on a larger number of less-powerful machines gives advantages in both performance and reliability.

### Variation 1: Medium Users, Medium Content

*Table 3-5: Four-Machine Configuration, Variation 1*

Machine	Components
1 (portal Web application components)	<ul style="list-style-type: none"><li>• Portal Server</li><li>• Image Server</li></ul>
2 (additional Web application components)	<ul style="list-style-type: none"><li>• WS Server</li><li>• Collaboration Server</li><li>• Content Server</li><li>• Studio Server</li><li>• Analytics Server</li><li>• Portlets</li></ul>
3 (search and indexing components)	<ul style="list-style-type: none"><li>• Automation Server</li><li>• Search Server</li></ul>
4 (persistent storage components)	<ul style="list-style-type: none"><li>• Document Repository</li><li>• Portal, Collaboration Server, Content Server, Workflow, Studio Server, and Analytics Server Databases</li></ul>

### *Notes and Guidelines*

This is a reasonable, general purpose configuration.

- Separating the portal server from the portlets will help reliability, particularly if the portlets connect to an unreliable back-end or require other software to be installed.
- Combining the Automation Server and the Search Server is not ideal, because indexing activity taxes both of those components heavily. Indexing will therefore take longer than it otherwise might and should be scheduled at night if possible.
- This system is not optimized for availability. If the Portal Server goes down, the system will not be available for users. If availability is a higher priority or if you expect a large number of users, consider Variation 2.
- If the amount of content is high, the front-end components can be combined onto a single machine. This allows separation of the Automation Server and Search Server onto separate machines. The expectation is that crawling activity on the Automation Server will be high and that indexing activity on the Search Server will be high as well.

Variation 2: Higher Availability, More Users, Less Content

Table 3-6: Four-Machine Configuration

Machine	Components
1 (portal Web application components)	<ul style="list-style-type: none"><li>• Portal Server</li><li>• Image Server</li></ul>
2 (portal Web application components, load balanced with Machine 1)	<ul style="list-style-type: none"><li>• Portal Server</li><li>• Image Server</li><li>• WS Server</li></ul>
3 (additional Web application components)	<ul style="list-style-type: none"><li>• WS Server</li><li>• Collaboration Server</li><li>• Content Server</li><li>• Studio Server</li><li>• Analytics Server</li><li>• Portlets</li></ul>
4 (back-end service components)	<ul style="list-style-type: none"><li>• Automation Server</li><li>• Search Server</li><li>• Document Repository</li><li>• Portal, Collaboration Server, Content Server, Studio Server, and Analytics Server Databases</li></ul>

### *Notes and Guidelines*

This configuration makes sense when reliability and availability is more important than performance.

- This system is optimized for availability and reliability of the Portal Server. Separating components from the Portal Server minimizes the chances that it will go down. In the event that one server does go down, it is load-balanced with a twin so the system is still available to end-users.
- This system is not ideal if there is a large amount of content in the Knowledge Directory or if that content needs to be refreshed often. Jobs on the Automation Server should run in the off hours if possible. If content and search are more important, consider Variation 3.

Variation 3: Large Amounts of Knowledge Directory Content

Table 3-7: Four-Machine Configuration, Variation 3

Machine	Components
1 (Web application components)	<ul style="list-style-type: none"><li>• Portal Server</li><li>• Image Server</li><li>• WS Server</li><li>• Collaboration Server</li><li>• Content Server</li><li>• Studio Server</li><li>• Analytics Server</li><li>• Portlets</li></ul>
2 (crawling components)	<ul style="list-style-type: none"><li>• Automation Server</li><li>• Crawler Web Services</li></ul>
3 (indexing and search components)	<ul style="list-style-type: none"><li>• Search Server</li></ul>
4 (persistent storage components)	<ul style="list-style-type: none"><li>• Document Repository</li><li>• Portal, Collaboration Server, Content Server, Studio Server, and Analytics Server Databases</li></ul>

### *Notes and Guidelines*

This configuration is optimized for crawling, indexing, and searching content.

- Crawling and indexing could take place at any time without a large impact on end-users.
- In this case, the Search Server should have a large amount of RAM for best performance.
- This system would be less reliable and available than the previous configuration, since there is only one Portal Server and it shares a computer with other components, particularly portlets. If the portlets require additional software installation or are CPU intensive (e-mail portlets, for example), consider adding an additional computer.



**Important:** For many four-machine deployments, and almost every deployment using an Oracle database, IT best practices might dictate that the database server be hosted, managed, and sized separately from all other portal components. In this case, you should use the three-machine configuration, using the fourth machine for the database and placing the Document Repository on the same machine as other back-end services like the Search and Automation Servers.



## Five-Machine (or More) Configuration

Going beyond a four-machine configuration introduces more potential variations. The most important metrics are the number of users and the amount of content. Typically the number of users has a much greater impact on horizontal scaling options for the Portal Server than the amount of content.

With five machines or more, if you are seeking to maximize reliability you will likely choose a larger number of smaller computers. General rules when seeking to maximize reliability are:

- Load balance Portal Servers first.
- Separate portlets that access other back ends, particularly those that require non-Plumtree software to be installed on the Portal Server.
- If collaboration is an important part of your deployment, separate the Collaboration Server from other components next. You can also load balance Collaboration Servers.
- If your deployment has a great deal of content, separate the Automation Server from the Search Server.

This section describes how additional servers of different types affect your deployment:

- **Additional Portal Servers** - Adding additional machines to host the portal Web application increases the capacity for concurrent users, as well as the reliability of the overall system. Portal Servers can be placed behind a third-party load balancing system.
- **Additional Automation Servers** - Adding additional machines to host Automation Servers increases the ability to run multiple jobs that crawl content, import user profiles, or synchronize security with other repositories and identity management systems. However, the Search Server cannot be horizontally scaled. As a result, adding more than three or four Automation Servers for the purpose of crawling content does not improve system performance, since indexing activity is ultimately bottlenecked by the Search Server.
- **Additional Remote Servers** - Adding additional machines to host portlets increases the capacity for concurrent users, similar to using additional machines for Portal Servers. Because remote servers typically interact with third-party software to provide specific web services, the desired number of machines can vary widely from one deployment to

another. In general, separating portlets onto their own server is most necessary when the portlets interact with back-end systems whose reliability is variable, and when the portlets must be timely and therefore cannot take advantage of caching on the Portal Server.

- **Additional Collaboration Servers** - Adding additional machines to host Collaboration Servers increases the capacity for users working with collaborative projects, tasks, discussions, and documents. Collaboration Servers can be placed behind a third-party load balancing system. If DNS aliases are established, multiple instances of the Collaboration Server can be load balanced using the portal's Parallel Portal Engine (PPE).
- **Additional Search Servers** - It is possible to horizontally scale query functionality (as opposed to indexing) by having separate query and index servers and using search replication to synchronize indexed data. Multiple query servers can then be configured behind third-party load balancing hardware. Because of the increased administrative costs, it is important to carefully analyze the benefits and risks of such an advanced deployment. Contact the Plumtree Professional Services organization for details.
- **Additional Content Servers** - It is not possible to horizontally scale Content Server by adding additional machines. Only one Content Server instance can access the Content Server Database at a time. It is important to note that Content Server portlets have a primarily administrative function, and the actual published content is distributed to a file system or FTP site. The published content can then be delivered by a Web server, which can be horizontally scaled through conventional means. It is possible to install more than one Content Server if each has its own database and folder in the administrative object directory for portlets. Because of the increased administrative costs, it is important to carefully analyze the benefits and risks of this type of deployment. Contact the Plumtree Professional Services organization for details.
- **Additional Studio Servers** - It is not possible to horizontally scale Studio Server by adding additional machines. Only one Studio Server instance can access the Studio Server Database at a time. However, it is possible to install more than one Studio Server if each has its own database and folder in the administrative object directory for portlets. Because of the increased administrative costs, it is important to carefully analyze the benefits and risks of this type of deployment. Contact the Plumtree Professional Services organization for details.

- **Additional Analytics Servers** - It is not possible to horizontally scale Analytics Server by adding additional installations. Only one Analytics Server instance can access the Analytics Server database at a time.

## Hardware Sizing and Scaling

This section provides guidelines for estimating initial, optimal capacity and configuration requirements for Plumtree components. This section assumes that you understand your business requirements and your network operating environments. Consult with Plumtree's Professional Services Organization to tailor a configuration that makes sense for your specific requirements.

As we have seen, the Plumtree Enterprise Web Suite is made up of a number of different components that can reside on separate computers. While all of these components have their own sizing and scaling characteristics, the Portal Server is a component through which end-users access all other functions of the suite. Understanding how the Portal Server scales, and how much capacity you need, is thus the first task in planning a Plumtree Enterprise Web Suite deployment.

### Portal Server Sizing and Scaling

It is often assumed that portal deployment scaling depends directly on the number of concurrent users or the total number of portal registered users. In fact, this is not the case. The term concurrent user is a legacy term from client/server systems, and Web-based systems do not have heavyweight connections as client/server applications do. HTTP 1.1 keep-alive connections and server-side state are used, but they use resources (TCP/IP sockets and memory) that are not constrained under ordinary circumstances. Resources such as file handles, database connections, and HTTP connections, are pooled for re-use by any user. Similarly, the number of registered users in a portal is also not informative. For the purposes of portal scaling, having many registered users means that a database table is slightly longer, but the size of the database table does not affect most portal operations significantly.

*The Portal Server has performance and scalability characteristics dependent directly on the page requests (or “hits”) per second the Web servers receive and the usage pattern (distribution of page hits across different portal features).* The limiting factor for Portal Server scalability is CPU resources on the computer hosting it. More CPUs on the Portal Server, and more Portal Servers in a Web farm, allow the portal to handle more page requests. The Portal Server scales linearly with respect to additional computers in the server farm—two computers are roughly twice as good as one. In Windows deployments, scalability with respect to additional CPUs in a single computer is not linear—expect roughly a sixty-five percent throughput increase from going from one CPU to two, and a similar increase from two CPUs to four. In Unix deployments with properly tuned JVMs, the Portal Server achieves near-linear multiprocessor scalability up to four CPUs.

While more users usually generate more hits, estimating the number of hits per user per amount of time before the deployment begins usually involves a certain amount of guesswork. In addition, the mix of activities those users perform also impacts the overall performance and scalability of the system, but it is difficult to know a priori what sorts of actions users will perform. Plumtree therefore cannot guarantee that the recommendations made here will hold for your deployment. We provide numbers of hits per second that companies can reasonably expect from given hardware, but these numbers can appear more precise than they actually are, and we recommend that companies adopt a coarse-grained approach to Portal Server scalability. Fortunately, modern server hardware is typically powerful enough that this approach is adequate for the purpose of scaling a portal deployment.

## Estimating Portal Server Traffic

To estimate the expected load on a Portal Server, extrapolate existing data based on known usage of a site that serves a similar function. If this is not possible, you will need to make a theoretical projection based on assumptions.

If possible, split the user population into groups based on usage-power, normal, and infrequent users. It is important to split the population into usage groups because a user hitting the system ten times per hour exacts ten times as much overhead on the system as a user who hits it once per hour. Concurrency, in the sense of maintaining an active session, is much less important than activity level.



**Note:** A power user is a user who frequently uses the portal and/or a user who performs activities that consume a lot of system resources. Adding or deleting portal content, for example, puts a higher load on the system than simply viewing content.

Plumtree recommends that hardware requirements be based on peak use of the Portal Server. If the Portal Server must be available to all users at all times, it must be available to all users during the peak time. To find the peak load, you must know how many users can be connected at one time. When the total number of each type of user is determined, multiply the number of users by their average hit rate (hits/second/user) and add the resulting hit rates (hits/second) to get the total number of hits/second on the Portal Servers. This is an average for peak load, depending on your load requirements. Use this total hit rate to determine Portal Server hardware. Use the following formula:

$$\text{number of hits/second} = ((\text{Power user hits/hour} * \text{\#power users} + \text{Normal user hits/hour} * \text{\#normal users} + \text{Infrequent user hits/hour} * \text{\#infrequent users}) \text{ hits/hour}) / (3600 \text{ seconds/hour}) * \text{fraction of users who could log on who are actually connected}$$

Here is an example calculation. Assume that during peak use of an employee portal:

- A power user hits the portal 12 times per hour, and 5% of users are power users.
- A normal user hits the portal 8 times per hour, and 25% of users are normal users.
- An infrequent user hits the portal 2 times per hour, and 70% of users are infrequent users.

Assume that there are 10,000 users, and that they are clustered heavily in headquarters' time zone, so that between 9:00 AM and 10:00 AM, 50% of the users hit the system.

Using the above formula, we get:

$$\text{number of hits/second} = ((12 * 500 + 8 * 2,500 + 2 * 7,000) \text{ hits/hour}) / (3600 \text{ seconds/hour}) * 0.5$$

This yields 5.5 hits/second

Now, change usage assumptions to see how they affect the calculation. We will continue to assume 10,000 users, half of whom are using the system at peak times, but we will increase the amount of usage:

- A power user hits the portal 20 times per hour, and 25% of users are power users.
- A normal user hits the portal 12 times per hour, and 50% of users are normal users.
- An infrequent user hits the portal 5 times per hour, and 25% of users are infrequent users.

These assumptions yield a rate of 17 hits/second.

Clearly, estimating the amount of usage correctly is important to sizing a deployment precisely, and there is a significant margin for error. Fortunately, as we shall see, a company who load balances two modern 2-CPU Pentium 4 Portal Servers would probably have excess capacity even under the more demanding usage assumptions and even if one of the Portal Servers were offline for maintenance.

## Benchmark Data for Portal Server Capacity in a Windows Deployment

Table 3-8 on page 3-37 provides guidelines for the number of hits per second you might expect from a Portal Server running with the designated hardware. All numbers are approximate and based on the following assumptions:

- The latest version of the Plumtree Corporate Portal is running.
- The installation uses remote portlets that cache intelligently. Typical usage patterns concentrate on the My Page and community page and are significantly impacted by the number of portlets, use of communities, and community portlets. For this benchmark we assume 10 portlets per page.
- Approximately 75% of the page requests are for My Pages and community pages, with the remainder made up of Knowledge Directory browsing and searches.
- The configuration includes a separate Administrative Portal Server. Administrative functions can put a greater and less predictable strain on Portal Servers than either portlets or Knowledge Directory hits.
- The computer hosting the Portal Server hosts no other components.
- The installation uses a separate Image Server.

- The installation is appropriately configured beyond the Portal Server. The database server and Search Server are hosted separately on hardware that does not bottleneck the deployment. There are no network latencies between any of the components.
- The installation uses SQL Server.
- The Portal Servers have 1 GB of RAM.
- The values represent nominal load. That is, they represent a sustainable level of strain on the Portal Server computer, where CPU usage does not exceed 75% of the capacity of the server. Plumtree does not recommend running servers at more than 75% of their CPU capacity for sustained periods.

*Table 3-8: Hits/Second at Nominal Load*

<b>System Details</b>	<b>Hits/Second at Nominal Load</b>
4x700 Pentium 3 Xeon, 1 MB Cache	30
4x550 Pentium 3 Xeon, 1 MB Cache	23
2x550 Pentium 3 Xeon, 1 MB Cache	14
2x700 Pentium 3, 256K Cache	14
1x700 Pentium 3, 256K Cache	9
2x1.4 Pentium 3, 512K Cache	30
2x1.8 Pentium 4, Xeon 512K Cache	31
As above, with Hyperthreading enabled	36
2x2.8 Pentium 4 Xeon 512K Cache	42
As above, with Hyperthreading enabled	48



**Important:** Microsoft .NET, WebSphere, WebLogic, and Tomcat have very similar overall performance characteristics. Performance characteristics for the application servers differ somewhat from page to page, but not enough to affect the aggregate throughput more than 5% or so. Note, however, the following:

- WebSphere requires more memory than the other application servers. Companies should not run a production Portal Server using WebSphere with less than 2 GB of memory.
- Both WebSphere and Tomcat will require tuning to attain optimal performance. WebLogic and .NET perform very well using their default settings. Consult the Plumtree Knowledge Base for the latest information on tuning WebSphere and Tomcat.
- Under high load, Tomcat refuses browser connections for approximately 1 in 10,000 requests. Users who experience this problem just need to refresh the page or follow the link again. However, load tests might experience these errors. Tuning can affect this error rate (decreasing or increasing it); again, consult the Plumtree Knowledge Base for the latest information on tuning Tomcat.
- If Tomcat is used in production, Plumtree strongly recommends placing it behind an HTTP Server such as Apache. Companies might also experience stability issues with Tomcat under SSL at high load.

In planning the capacity of your server, make the following adjustments:

- If users use My Pages more than communities, revise the number upward by approximately 5%.
- If you have a Windows deployment that includes Collaboration Server and it is heavily used, revise the number upward by 10%. (The same does not apply to Unix deployments.)
- If users use the Knowledge Directory more than 20% of the time, revise the number downward by approximately 10%.
- If the deployment uses Oracle rather than SQL Server, revise the number downward by 20%.



- If the deployment runs under SSL (security mode 2) on the Portal Server, without an SSL accelerator, revise the number downward by 25%.
- If the deployment does not have a separate Image Server, revise the number downward by 5%.
- HTTP compression utilities such as PipeBoost (IIS) or mod\_gzip (Apache) decrease throughput roughly 10%, but can also decrease the negative impact of SSL by a factor of 8, therefore improving performance if SSL is simultaneously used and not offloaded.
- If this Portal Server also serves administrators, revise the number downward by 15%. If the same computer will serve administrators and end-users, Plumtree recommends using a multi-CPU machine.
- Using a virus scanner on the Portal Server will degrade performance 5-15%, and might prevent the portal from functioning properly.

Generally, more processor cache improves performance. Approximately 20% performance improvement can be achieved for 1 MB of cache versus 256 KB (Pentium 3 Xeon 1 MB versus Pentium 3 with 256 KB cache), and 2 MB cache provides another 5-10% in system throughput. However, these are rough guidelines. The performance boost that larger or faster cache gives is dependent on processor speed and the number of processors. Computers with more than four processors would get the most performance boost from the largest cache possible. With four processors or fewer, the price/performance ratio varies widely, and more careful evaluation is necessary.

Most importantly, fast caches and memory subsystems are important to portal performance. Database and Search Server performance are also very cache size/speed dependent, while Automation Server performance is less cache dependent.

If you have a Windows deployment, additional memory improves performance, but only up to about 4 GB of memory unless the deployment is very large and using a large portlet cache. (The same does not apply to Unix deployments.)

## Benchmark Data for Portal Server Capacity in Unix Deployments

The following table provides guidelines for the throughput you might expect from a Portal Server running on Solaris with the designated hardware. All numbers are approximate and based on the following test configuration:

- Approximately 75% of the page requests were for My Pages and community pages with 10 portlets per page. The remaining requests were made up of Knowledge Directory browsing and searches. The number of portlets per page and the use of communities and community portlets have a significant impact on the throughput.
- There were dedicated computers for the Portal Server, Image Server, Search Server, and database server.
- All components downstream from the Portal Server were hosted on hardware that did not bottleneck the deployment.
- The database was SQL Server.
- These values represent a sustainable level of throughput on the Portal Server computer, where CPU usage does not exceed 75% of the capacity.

Table 3-9: Portal 5.0.3J/Solaris 9 Throughput at Sustainable Load

Portal Server Hardware	Pages/Second at Sustainable Load
1 x UltraSparc II 450Mhz	9
2 x UltraSparc II 450Mhz	18
4 x UltraSparc II 450Mhz	35
1 x UltraSparc III 1Ghz	18
2 x UltraSparc III 1Ghz	36
4 x UltraSparc III 1Ghz	70

### Portal Server Capacity and Availability

As we have seen, a company with 10,000 users might only generate five hits per second on the Portal Server machine. That company might consider a deployment with a single 2-CPU Pentium 4 computer, which should, even under SSL, provide enough CPU power for five times that many hits. However, capacity is not the only factor, or even the primary factor, in most deployments. Companies should strive for a deployment that has both the capacity to

serve the expected user community and redundancy to handle unexpected malfunctions and scheduled maintenance.

The Portal Server computer must be up and running smoothly at all times because it faces the end-user population directly. To ensure predictable, reliable Portal Server performance, Plumtree recommends that the Portal Server component be hosted on computers separate from other Plumtree components. Such a configuration is more secure since persistent data (search and database) and back-end tasks (Automation Server) are not on the same computer.

*Load balancing multiple Portal Servers further increases reliability and is the primary reason for using multiple Portal Servers, even if one computer provides enough capacity to meet your deployment needs.* If one computer goes down, requests fail over to the other machines in the Web farm. Additionally load balancing allows you to install software patches and upgrades without bringing down the system, since one computer at a time is taken out of the rotation. The more lower-powered machines used in the Web farm, the higher the potential availability. Plumtree recommends a compromise between availability and having fewer machines to manage and maintain. Two 2-CPU computers are often cheaper than one 4-CPU computer. They provide a much better price/performance ratio for Portal Servers, and will improve availability.

*Thus, Plumtree strongly recommends that you begin your deployment with two Portal Server computers load balanced.* After you have set up load balancing, the system is easy to expand.

When considering availability, determine what fraction of the expected peak load each computer should be able to serve. Should one server go down, the capacity of the system is reduced by that fraction. Ideally, the remaining computers should still be able to handle a peak level of traffic. If availability is a high priority, determine the number of extra servers necessary for fail-over, not only in the load balanced farm, but also those able to be swapped in, in case of failure. This number balances the cost of high-availability hardware features per server, such as redundant hard drive systems and power supplies. Also consider using older or less expensive servers as fail-over machines.

The Portal Server scales horizontally. This means that the load capacity of a Portal Server Web farm grows linearly with the number of computers in the farm. For example, if you determine that a capacity of 100 hits per second is required, and you have a computer that

will provide 25 hits per second, four instances of that computer will provide approximately 100 hits per second.

Because the Plumtree system can easily scale out to multiple Web servers, it is very easy to expand the portal later by adding more Web server computers. When the system has two or three Web server computers, it can also be advantageous to have a separate Image Server computer.

## Database Sizing and Scaling

If at all possible, the database should be on a separate computer. As the system grows, the database will need its own computer to have enough resources, and moving a database is an expensive operation. Some third-party components installed on other Plumtree server computers can adversely affect the networking on the database computer. If the database goes down, the entire system goes down. Tuning the database is difficult if it shares a computer with other components, and tuning operations can interfere with the operation of other components.

The details of database performance and reliability are best left up to a good DBA. The minimum hardware for a reliable database is a fault-tolerant disk subsystem, lots of ECC RAM, and enough CPU and network bandwidth to fill the needs of your Portal Servers and Automation Servers. The CPU load on a portal database is significantly less than that on the Portal Servers, but the size and speed of the processor L2 or L3 cache has a very important effect on database performance. All databases should be regularly backed up, and back-ups should be stored in a safe place and archived off-site for disaster recovery.

A rule of thumb for database CPU power is that the database utilizes approximately 1/8 to 1/4 of the resources of a Portal Server or Automation Server running at full capacity. That is, if all the Portal and Automation Servers are the same 2-processor 700 MHz configuration, then one 2-processor 700 MHz database system can support between four and eight Portal Servers and Automation Servers. Because of the high variability in usage patterns, especially on Automation Servers, Plumtree recommends having excess capacity on your database server. The database is central to the whole system so it is more important to keep it up all the time and make it easy to expand. Therefore, we recommend approximately a 1:4 ratio for Portal Servers and a 1:2 ratio for Automation Servers rather than 1:8,

keeping the database well below peak capacity and leaving room to expand. That is, the CPU of the database should be approximately:

$(\frac{1}{4} \text{ the CPU for all the Portal Servers combined}) + (\frac{1}{2} \text{ the CPU of all the Automation Servers combined})$

This represents a very conservative estimate that leaves plenty of headroom. If there are plans for future expansion, it is probably worth investing in a database that can handle the projected load after expansion. To allow for portal growth, the database, initially, should be at least twice as powerful as required.

Databases are usually memory gated. If the server has at least 2 GB of memory, it is usually fine. Very large databases (1,000,000 users or a combination of fewer users but a very large number of portlets or documents) might require 4 GB of memory. With this configuration, the database should all fit in memory, eliminating most I/O operations (which are slow and can block other operations).

Plumtree recommends the use of RAID disk arrays. Since only a single database (or cluster of databases) can be used, reliable data storage is critical. Portal databases are relatively small. For example, a database for a portal of 1,000,000 users probably will not exceed 5-10 GB in size. A portal with 25,000 users and 50,000 documents will be approximately 500 MB in size. For appropriate RAID levels, consult your DBA or the database vendor.

The relational database uses memory to cache tables, so adding memory to the database computer improves performance. Larger installations should be served from multi-CPU databases.

## Search Server Sizing and Scaling

Plumtree generally recommends a 4-CPU Windows Advanced Server system with 4 GB of RAM and 3 GB of virtual memory enabled. Given the importance of sufficient RAM for good Search Server performance and the affordability of memory, such a system is a good choice for most deployments. RAID storage can improve performance. Plumtree recommends software-controlled RAID as an affordable storage option. Exceptions to this recommendation include deployments with very large document collections and scenarios where there are serious budgetary constraints.

In cases where the cost of the Search Server machine needs to be minimized, in order to determine the amount of RAM required for the Search Server, you should estimate the total amount of raw text that will be indexed. At most, this will be the total size on disk of all the documents indexed. Of course, for some document types, such as Microsoft Word or Adobe PDF, the amount of raw text is considerably less than the size of the file on disk (for Word and PDF the amount of text is typically 1/3 or so of the file size). Other document types, such as .txt and .html, are almost entirely raw text. For example, if the Search Server indexes 100,000 documents which average 30 KB and are all of type .txt, then the amount of raw text will be roughly 3 GB, whereas if the docs are all Word docs, then a rough estimate would be 1 GB. In most real deployments, a mixture of different document types will be indexed, and the estimate of raw text will need to be adjusted accordingly.

Once an estimate of the amount of raw text indexed has been calculated, an appropriate amount of RAM for the Search Server machine can be determined. Sufficient RAM is critical to ensure good Search Server performance. Again, if price considerations are not critical, Plumtree recommends the 4 GB system described previously. However, if there are serious budgetary constraints and a machine with less memory must be used, the following guidelines might be useful: A system with 1 GB of RAM should be sufficient for collections with up to 625 MB of raw text, while a system with 2 GB of RAM should be able to handle up to 3.75 GB of raw text. Be sure to use the win32-medium.ini configuration file for a 1 GB system and the win32-large.ini configuration file for the 2 GB system. Search Server configuration files are described in [“Search Server Configuration File Sizing” on page 3-46](#).

For very large collections (for example, those with more than 500,000 documents or more than 6.25 GB of raw text) a 64-bit Solaris Search Server should be considered. Solaris offers the best stability and scalability in terms of RAM expansion. For smaller collections, however, the cost of a Solaris server cannot be justified, and a Windows server should be sufficient.

CPU requirements for the Search Server are largely a function of the number of portal users. Generally speaking, the Search Server is highly scalable in terms of query speed. Plumtree has performed scale experiments with a test collection of 110,000 documents averaging 28 KB, for a total of 3.1 GB of raw text indexed. Our experiments indicate that a 4-CPU@700Mhz system is capable of handling up to 156,000 portal users, under reasonable assumptions about the frequency of usage of search (these assumptions are described in

[“General Assumptions About the Search Server Environment” on page 3-49](#)). A 2-CPU@700Mhz system would be sufficient for up to 78,000 users.

## Search Server Scaling and Performance

*In order of importance, the most important factors in Search Server performance are adequate RAM, processor speed, disk system including cache, and number of CPUs.*

The Search Server receives requests for indexing and searching from other portal components via network TCP connections. Indexing and querying are memory intensive, and the Search Server makes extensive use of memory mapping for efficient file I/O. As a result, it is extremely important to have at least 3 GB of paging file or swap space to support virtual memory operations. In addition, it is important to have an adequate amount of physical RAM to support internal caching done by the Search Server. (Internal Search Server caching is completely separate from file system or portal/portlet caching).

Because file I/O is such an important part of indexing and querying operations, the Search Server needs high-end disk hardware. Installing the Search Server on low-latency disk drives or disk arrays is important for performance. Any kind of fast, local disk is acceptable (for example, SCSI, RAID arrays), and there are no known restrictions with respect to hardware or OS disk caching. Backups should be performed using replication to provide a consistent snapshot of the search index files in the presence of ongoing indexing activity.

The Plumtree Search Server maintains a single directory containing the binary index files for the searchable documents and folders in the Knowledge Directory. There are no distinct files or subdirectories corresponding to properties or full-text.

The Search Server computer needs enough disk space to hold the entire text content of all indexed documents. Every 10,000 documents that use the same document type will occupy ~750 MB of disk space. Document types will occupy ~750 MB of disk space. If all documents link to large binary-format text-based files (Word, PDF), the search collection will usually be ~1/3 of the cumulative size of all the files. Formats with predominantly text content (.html, .txt) have a higher ratio of search collection disk space to cumulative data size, while large files with images (PowerPoint) require a much smaller ratio. Plumtree recommends allocating free disk space of at least 3 times the collection size with a minimum of 5 GB of free disk space. This allows for expansion and intermediate file creation that occurs

while the Search Server indexes new content. If there is not enough disk space during indexing, the Search Server enters a read-only state and will not index new content until additional space is made available and the Search Server is restarted.

To ensure best performance, the Search Server should have very fast, dedicated network connections to all Portal and Automation Servers. All communication between the Search Server and Portal or Automation Servers occurs over the network, via TCP connections. Network speed is rarely a limiting factor in system performance, however.

Search Server indexes content faster and requires less maintenance with additional memory. If the portal will index a great deal of content, additional memory on the Automation Server and Search Server computers is desirable.

## Search Server Configuration File Sizing

The Plumtree Search Server includes a set of example configuration files that include recommendations based on the amount of RAM available for the Search Server. Table 3-10 describes details of the Search Server cache parameters (the settings in the table are described next). However, Plumtree strongly recommends that you choose one of the example configuration files based on the available physical RAM in your machine, rather than attempting to experiment with the cache parameters individually.



**Note:** When co-locating the Search Server with other Plumtree components, be sure to reduce the memory footprint that the Search Server uses. In so doing, you effectively place it on a smaller machine.

To use the example file settings, save a copy of your existing **ignite.ini** file as “ignite\_original.ini”; then, save a copy of the desired example file as **ignite.ini**. (**ignite.ini** is located in the Search Server installation directory, for example, C:\Program Files\plumtree\ptsearchserver\config.)



*Table 3-10: Search Server example configuration files*

<b>File</b>	<b>RAM</b>	<b>Settings</b>
Win32-small.ini	512 MB	<ul style="list-style-type: none"> <li>• RF_DOCUMENT_TOKEN_CACHE_SIZE=250000</li> <li>• RF_SPELL_TOKEN_CACHE_SIZE=250000</li> <li>• # Index cache: 75 MB</li> <li>• RF_INDEX_CACHE_BYTES=78643200</li> <li>• # Docset cache: 25 MB</li> <li>• RF_DOCSET_CACHE_BYTES=26214400</li> </ul>
Win32-medium.ini	1 GB	<ul style="list-style-type: none"> <li>• RF_DOCUMENT_TOKEN_CACHE_SIZE=1000000</li> <li>• RF_SPELL_TOKEN_CACHE_SIZE=500000</li> <li>• # Index cache: 75 MB</li> <li>• RF_INDEX_CACHE_BYTES=78643200</li> <li>• # Docset cache: 25 MB</li> <li>• RF_DOCSET_CACHE_BYTES=26214400</li> </ul>
Win32-large.ini	2 GB	<ul style="list-style-type: none"> <li>• RF_DOCUMENT_TOKEN_CACHE_SIZE=1000000</li> <li>• RF_SPELL_TOKEN_CACHE_SIZE=500000</li> <li>• # Index cache: 450 MB</li> <li>• RF_INDEX_CACHE_BYTES=471859200</li> <li>• # Docset cache: 150 MB</li> <li>• RF_DOCSET_CACHE_BYTES=157286400</li> </ul>
Win32-maxium.ini	4 GB with 3 GB switch running on Windows 2000 Advanced Server	<ul style="list-style-type: none"> <li>• RF_DOCUMENT_TOKEN_CACHE_SIZE=1000000</li> <li>• RF_SPELL_TOKEN_CACHE_SIZE=500000</li> <li>• # Index cache 1 GB</li> <li>• RF_INDEX_CACHE_BYTES=1048576000</li> <li>• # Docset cache 334 MB</li> <li>• RF_DOCSET_CACHE_BYTES=350224384</li> </ul>

## Configuration File Settings

This section describes the settings listed in the table. The Search Server configuration file includes other settings, which are described in the *Administrator Guide for the Plumtree Corporate Portal*.

- **RF\_DOCUMENT\_TOKEN\_CACHE\_SIZE:** the default value is 250000. This parameter has a significant effect on Search Server indexing and query performance, with larger values providing better performance. In practice, values larger than 1000000 provide diminishing return while consuming significant amounts of memory. Each cache element is 120 bytes in size, so the default document token cache will occupy 29 MB of memory.
- **RF\_SPELL\_TOKEN\_CACHE\_SIZE:** the default value is 250000. This parameter has a significant effect on Search Server indexing performance and the performance of queries involving spell-checking, spell-correction, or wildcard operations. Larger values provide better performance. In practice, values larger than 1000000 provide diminishing return while consuming significant amounts of memory. Each cache element is 120 bytes in size, so the default spell cache will occupy 29 MB of memory.
- **RF\_INDEX\_CACHE\_BYTES:** numeric parameter specifying the size of the index cache in bytes. The default value is 78643200 (75 MB). The value of this parameter has a significant effect on Search Server query performance.
- **RF\_DOCSET\_CACHE\_BYTES:** numeric parameter specifying the size of the document cache in bytes. The default value is 2614400 (25 MB). The value of this parameter has a significant effect on Search Server query performance.

The index and docset caches should be made as large as possible while leaving sufficient memory available for the Search Server's other needs.

## General Assumptions About the Search Server Environment

General assumption of the environment:

- Users refresh portal pages 8 times per hour or one hit in every 480 seconds.
- Ten percent of portal users use search at any given time.
- In a regular hour, the number of concurrent portal users is less than 3% of the total number of portal users
- The peak for the number of concurrent portal users is about 30% of the total number of portal users.
- The number of concurrent search users is about 3% of the total users in peak times.
- Large indexing requests occur off-hours. During peak times most searches are from the portal banner. Few search update jobs run during the normal business hours.
- There is an equal probability of different type of search queries: single term, phrase, and wildcard.

Under these assumptions, portal users request a search an average of once every 480 seconds. At peak time, portal banner search is requested at a rate of  $(\text{total \# of users} * 0.3 * 0.0021)/\text{sec}$ . In a regular business hour, the portal banner search is requested an order of magnitude less frequently than at peak time.

## Administrative Portal Server Sizing and Scaling

Computers running Administrative Portal Servers or Automation Servers are more likely to experience problems than those running Portal Servers or the database because administrative operations are more unpredictable, use more system resources, and do more writes than reads. Any components sharing a computer with either the Administrative Portal Server or the Automation Server might feel the impact unpredictably because of the variable resource usage of these components. For example, an administrative user might delete a folder containing 10,000 users, a resource intensive task that might affect other administrative users.

Even for large installations, there are generally not a large number of content managers and portal administrators. One administrative computer is usually enough. If high availability of the Administrative Portal Server is not a high priority and there is a portal administrator on

site to deal with difficulties, then it is probably acceptable to have one Automation Server run on the same computer as the Administrative Portal Server. Although there are fewer users capable of using the Administrative Portal Server than the Portal Server, these users can each perform very resource-intensive operations; it is a good idea to have a server-class computer for the Administrative Portal Server, comparable to the Portal Server or Automation Server computers for scaled systems. If availability is a priority, then two, less-powerful computers can be load balanced to achieve the same capacity while providing flexibility for maintenance. It is usually advantageous to run the Administrative Portal Server inside a firewall; it is only accessible from the intranet and is more secure.

## Automation Server Sizing and Scaling

Automation Server processor use can be limited by pinning the Automation Server process to a particular processor, set of processors, or decreasing the priority of the process, but it is generally best to run the Automation Server on its own computer where it can use resources at will, yielding predictable and consistent job run times.

Automation Servers are responsible for a great deal of processing. In particular, they do user and group imports and synchronizations, crawls, publishing, link checks, and maintenance. The Card Refresh Agent (part of the set of maintenance jobs) is responsible for keeping links up to date with the source documents and is run multiple times in parallel to increase throughput. To determine the number and power of Automation Server computers, decide how often the various jobs must run and how much work each will have to do.

For example, if there are a large number of users to synchronize and you want to synchronize frequently to keep users up-to-date, then one Automation Server computer will run authentication source jobs all the time, and there will be much fewer resources to run other jobs. Further, if this is a priority operation for you, to ensure security consistency, you might want a dedicated Automation Server (or perhaps two for redundancy) that runs only this one job.

Automation Servers are fundamentally limited by processor speed and I/O, and do not benefit from large processor caches as much as Portal Server computers. If you have a Windows deployment, Automation Servers are also limited by RAM. Since I/O, and particularly network I/O, often blocks Automation Server performance, running the Automation Server

on a high-end server computer with multiple processors might not be worth the cost. Multiple single- or dual-processor computers are more cost effective.

Each crawl uses a base of approximately 60 MB of memory. In general, increasing memory allows it to handle a larger number of concurrently running jobs, whereas additional processors help speed up the processing time for each individual job. As a baseline, a single CPU computer with 256 MB of memory should be able to handle two concurrent crawls. Plumtree recommends two CPUs and 1 GB of memory for an Automation Server computer for medium or larger deployments. Also, the Virtual Memory paging file can be extended, allowing the computer to run more operations concurrently, without running out of physical memory.

RAM is especially important when rebuilding search collections. If the Plumtree system has more than 10,000 documents, rebuilding the search collection requires at least 256 MB of physical RAM and 500 MB of virtual RAM on the Automation Server computer. Collections with 50,000+ documents require 512 MB of physical RAM and 1 GB of virtual RAM.

As a rough estimate, allocate a powerful Automation Server for every 125,000 documents you plan to crawl into the Knowledge Directory. The more often document refresh and document changes take place, the more Automation Server power is necessary. Monitor actual Automation Server performance and add Automation Server computers as needed. The recommended maximum number of jobs an Automation Server can run simultaneously is four. However, very fast new hardware or large multi-CPU machines can handle more concurrent jobs, provided there is enough network bandwidth, processor power, RAM, and temporary disk I/O space. If the Automation Servers are fully utilized, consider adding another Automation Server. For large search collections, Plumtree recommends an Automation Server dedicated solely to the administrative operations: card refresh, search update, and user synchronization.

Crawled files are temporarily stored on the Automation Server in batches while they are indexed or written into the search collection. The temporary files are then automatically deleted. Since it is difficult to determine the size of the files to be indexed, it is best to allocate at least 2 GB of free disk space at all times for temporary file storage.

RAID arrays are not always necessary. Consider using two physical disks: one for the operating system and one for portal software. Disk performance is not a major factor in Automation Server performance.

Jobs can be load balanced across multiple Automation Servers by registering administrative folders containing jobs with multiple Automation Servers. If Automation Server availability is a requirement, an extra Automation Server can be used for load balancing to provide very high availability.

Estimate what the job schedule will be to keep portal users synchronized at the required frequency, the Knowledge Directory refreshed and updated at the proper frequency, and any crawlers, snapshot queries, or link checks performed when needed. This job schedule can be used to predict processor and memory usage for all times on the Automation Servers. Crawling and synchronization are substantially slower on Oracle than on SQL Server. Using SQL Server rather than Oracle will decrease the number of Automation Servers required if processing time is limiting factor.

Automation Servers index content faster and require less maintenance with additional memory. If the portal will index a great deal of content, additional memory on the Automation and Search Server computers is desirable.

## Image Server Scaling

The Image Server serves images and other static files. This is a simple operation for any Web server. Publicly available benchmarks can be used to predict performance. Generally, a computer similar to a Portal Server computer can easily handle the image load for all Web servers. Single processor systems with 512 MB of memory are more than enough in most situations.

The Image Server is limited by network I/O. It only needs enough RAM to cache all of the Plumtree images (or custom images) and static Web files along with enough RAM for the Web server and the operating system. This is seldom more than 256 MB. Today's processors can easily max out a 100baseT connection with flat file transfer of images.

Consider purchasing two physical disks: one for the operating system and one for the images.

## Remote Server Sizing and Scaling

The remote server hosts external web services—authentication web services, crawler web services, profile web services, portlet web services, and search web services—for the portal. It should be on a computer separate from other portal components. This permits the addition of more portlets; you can perform load balancing on the remote servers later. If you must put the remote server on the same computer as another component, put the remote server on the Administrative Portal Server or the Automation Server computer. Doing this can destabilize the portlets as well as the other component on the computer, but the Portal Server will remain running even if the remote server goes down. If taking portlets down periodically for regular maintenance and other problems is unacceptable, then you must put it on its own computer.

Hardware requirements for the remote server computer are similar to those of the Portal Server. However, the nature of the portlets or web services you use plays a big part in the choice of remote server computer. Accessing an unstable piece of back-end software by a portlet might cause the remote server to fail. You might be better off putting such a portlet, in its own process space or on its own remote server computer. Likewise, a portlet that heavily utilizes server resources might be a candidate for isolation on a separate server.

Depending on the types of portlets employed, the relative number and power of remote server computers to Portal Servers can vary greatly. Consult specific portlet scaling information to determine hardware requirements.

Remote Servers can be load balanced for scalability. For example, if 200,000 users access the e-mail portlet, you can load balance e-mail remote servers to handle the high load. The Plumtree Corporate Portal, with its Parallel Portal Engine built in, requires no extra hardware for load balancing remote servers.

## Custom Services Sizing and Scaling

This section provides guidelines for estimating requirements for the custom components you will deploy into your Plumtree Enterprise Web. This section assumes that you understand your business requirements and your network operating environments.

### Portlet Server

When you write a portlet using the Enterprise Web Development Kit (EDK), you are writing an application. Your application can perform almost any task, and depending on the task that it performs the hardware it requires will vary widely. The best way to conserve resources on a remote server running portlets is to start with a good design. The *Enterprise Web Development Guide* is filled with best practices, but some basics should always be remembered:

- Choose an appropriate caching strategy
- Avoid CPU intensive operations if the portlet cannot cache; minimize them even when the portlet is caching effectively
- Place static files, such as images and .js files, on the imageserver

Once you have written a portlet, you should run a load test simulating realistic use. Only a load test will characterize the hardware requirements of your application.

### Authentication Server

An authentication web service performs two distinct operations; (1) synchronizing users and groups and (2) authenticating users. Synchronizing users and groups runs as a job on a periodic basis. Authenticating users happens in real time, whenever a user logs in.

To ascertain the hardware requirements for an authentication web service (AWS), you must run a realistic synchronization and simulate a realistic authentication load, while monitoring the computer hosting the AWS. For the authentication load simulation, you should also monitor the Portal Server computer. If the AWS hardware is resource-limited, the length of the synchronization will increase, and the number of concurrent logins will be limited. It is possible to run the synchronization and authentication on different hardware if this is desired.



Factors which determine the length of a first-time synchronization include:

- Number of users
- Number of groups
- Number of group memberships
- Cost of query against user directory

The length of subsequent synchronizations will be determined primarily by the cost of the query and by the delta in users, groups, and group memberships.

In the first half of 2004 a testing tool will be released that will allow simulation of authentication load and synchronization runs without using a portal. These tools will be announced and made available on the Developer Center (<http://portal.plumtree.com>).

## Profile Web Service Server

A profile web service (PWS) is often run with an authentication web service (AWS), but is also often run alone. An AWS synchronizes and authenticates users; a PWS pulls in profile data about the users. You will probably set up a PWS for your user directory but also write some custom profile web services for custom applications that hold interesting information about your users.

To ascertain the hardware requirements for a profile web service (PWS), follow the guidelines for the AWS synchronization hardware requirements described previously.

In the first half of 2004 a testing tool will be released that will allow simulation of profile synchronization runs without using a portal. These tools will be announced and made available on the Developer Center (<http://portal.plumtree.com>).

## Search Web Service Server

A search web service (SWS) is executed when the user selects it and runs a search. The hardware requirements for a SWS depend entirely on (a) what the SWS is ultimately searching and what APIs are being used, and (b) the expected volume of search requests.

The only way to determine the true requirements of your search web service is to deploy it and simulate a typical load.

## Crawler Web Service Server

A crawler web service (CWS) performs one mandatory function and one optional function: it allows the Automation Server to asynchronously crawl information from a remote system, and it optionally governs the experience when a user clicks through to a link in the Knowledge Directory.

Plumtree has several crawlers available, and these are well tested for high load during crawl and during click-through. When you write your own crawlers, be sure to test their resultant load within your environment.

In the first half of 2004 a testing tool will be released that will allow simulation of crawler runs without using a portal. These tools will be announced and made available on the Developer Center (<http://portal.plumtree.com>).

## Using the Plumtree Remote Client (PRC)

The Plumtree Enterprise Web Development Kit (EDK) has a component called the Plumtree Remote Client (PRC): the PRC allows the developer to call back to the portal to do any number of things, including object queries and portlet creation. You can utilize the PRC from a portlet or profile web service, or from a stand-alone application.

The PRC interacts with the WS Server in the portal configuration. The PRC sends instructions in the form of SOAP messages to the WS Server, which are then executed.

The PRC exposes powerful functionality, but it is important to know that different queries and operations trigger different loads on the WS Server, the database server, and in some cases, the Search Server. Work with your portal administrator to figure out the performance impact your PRC code will have; the best way to do this is to monitor the system as you run your code.

There are some good rules of thumb to follow when using the PRC.

- If you are using the PRC from a portlet, cache the PRC response so you only make the SOAP round-trip when necessary. For a high-use portlet, do not call the PRC with every page-draw; the WS Server will not handle the load.

- With the initial releases of the PRC, database queries were available, but, with the January 2004 release of the EDK, new functionality has been added allowing certain queries to be done against the Search Server. Queries against the Search Server are, in general, more efficient than queries against the database.

## Collaboration Server Sizing and Scaling

The recommended hardware for deploying Collaboration Server depends on factors such as load, network throughput, and the types of operations that users are doing at any given moment. In general, writing to Collaboration Server is more expensive than reading from it, so users who collaborate actively are more taxing to the system than those who browse. While it is important to ensure that the Portal Server has headroom for peak periods, because it gates access to the entire Enterprise Web Suite, companies can scale Collaboration Server in a more approximate fashion, starting with one or two servers and scaling out only if significant performance problems arise.

In general Collaboration Server should scale well up to any number of simultaneous users and it is generally recommended to have at least one dedicated 2-CPU Collaboration Server machine per 1000 active users (meaning they upload documents and post messages to the Collaboration Server throughout the day) or up to 10,000 casual users (meaning they access Collaboration Server content occasionally during the day and do not usually modify or post new content). The machines should have at least 1 GB of RAM and should be deployed separately from other Plumtree components (for example the Document Repository or the portal) when possible.

## Content Server Sizing and Scaling

Content Server (along with the embedded workflow engine) is a Web application that provides administrative portlets for content creation and publishing. Content contributors enter data into forms to create content items, and those items are published to a location via FTP or file system copy after traversing a sequence of workflow stages. In a typical deployment, the number of users who contribute and approve content is a small fraction of the total number of registered portal users. Also, once the content is published, it is static, so that a large number of users can access HTML pages, files, and images published through

Content Server without much incurred processing overhead by the Content Server Web application itself.

So the main metrics for Content Server are:

- How many users will be contributing content on a regular basis?
- How many users will be accessing the final published content?

Content Server cannot be horizontally scaled—you can only have one Content Server Web application on a single machine corresponding to a Content Server database. The same is true for the embedded workflow engine. As a result, there are not as many deployment options as for the Portal Server or Collaboration Server.

If the first metric is high (that is, hundreds of users contribute content on a regular basis), then it will be important to scale the Content Server machine itself. The following are the scaling factors for Content Server, in order of importance:

- Use a dedicated machine
- Add more memory
- Use faster CPUs
- Use more CPUs

The first and most obvious step is to install Content Server (and workflow) on a dedicated machine. Content Server must be installed on the same machine as the embedded workflow engine.

The Content Server dynamically utilizes all available memory for caching purposes. In general, the more memory available, the better Content Server will perform. In addition to adding more memory to the computer, Content Server's Java memory settings must be adjusted, as well. A good rule of thumb is to set the -Xms and -Xmx settings to 80% of physical memory, assuming no other servers are running on the computer.

Finally, adding faster and/or additional CPUs to the Content Server machine will improve performance as well.

If the second metric is high (that is, thousands of users access the final published content), then a deployment should avoid using Content Server as a Web server itself to host the published content. Rather, content items should be published to one or more separate Web

servers (which can be running on separate machines), by setting the publishing targets for the various sections of content. With such an approach, the Content Server machine will not be a bottleneck for end-user access to content.

Although Content Server cannot be horizontally scaled, it is possible to deploy more than one complete Content Server instance (each with its own database) on a single portal. This can be an attractive option if there are multiple groups of content contributors working on completely separate content. For example, an IT group in charge of maintaining Web content for a set of externally-facing sites could use one instance of Content Server, while a second group that creates internally-facing news articles and employee updates could use a separate instance.

## Studio Server Sizing and Scaling

Because of its extensive use of in-memory caching, Studio Server should scale well with both a large number of concurrent users and a large number of Studio Server portlets being served. Consequently, perhaps the most significant variable that can limit performance is the amount of memory installed on the host computer, how much memory is allocated to the application server, and how Studio Server's caches are configured. In general, Plumtree recommends that the computer hosting Studio Server have at least 1 GB of RAM, with at least 500 MB allocated to the application server. Studio Server's caches should be configured with at least as many objects as are likely to be required based on expected usage. For details on how to configure Studio Server's built-in caches, refer to Plumtree Knowledge Base article 11211: "INFO: Changing the sizes of the Studio Server's built-in caches."

In most cases, a single 2-CPU computer should provide adequate capacity for Studio Server.

## Analytics Server Sizing and Scaling

The Analytics Server Remote Server host computer serves the graphical user-interface for the Analytics Server Console and portlets and requires a 1.6 GHz or higher processor, with 2 MB L2 cache, and 1 GB RAM. The sole scaling consideration is the storage requirements for the Analytics Server database. The baseline database size is what you can expect to have your database start out with as you begin collecting usage data for the Plumtree Portal. Baseline for an average Analytics Server database is 300 MB.

A baseline portal is defined as:

- 100 Communities
- 100,000 Documents
- 20,000 Users

You can compute your baseline based on the following growth estimates of your portal:

- 220 KB per additional 100 Communities
- 250 KB per additional 100 Documents
- 190 KB per additional 100 Users

The growth of the database is directly correlated to the number of page views present in the system. A page view is defined as a user visiting either a community or a my page.

We estimate 100 MB of growth per 1 million page views. You can anticipate the database growth to roughly follow these numbers.

*Table 3-11: Estimating Growth of the Analytics Server Database*

Average Daily Page View Traffic	Estimated Growth per Day
100,000	10 MB
500,000	50 MB
1,000,000	100 MB
2,000,000	200 MB
5,000,000	500 MB

Use the average daily page view traffic number to calculate your daily growth and extrapolate that to your monthly or yearly growth requirements. These numbers are calculated assuming no archiving of historical data.

For example, with a baseline portal that receives approximately 500,000 page views per day, you can extrapolate this to roughly 18.3 GB of storage per year.

If your portal receives roughly 5,000,000 page views per day, you can assume that your Analytics Server database will grow by roughly 183 GB per year.

## Scaling Using Federated Portals

One way of scaling a portal is to use multiple networked (federated) portals rather than one very large portal. This is especially useful if you require more than 25-50 GB of indexed content in your Knowledge Directory. It also makes sense if disparate departments need to share some data but use mostly different portlets, communities, and content. Sometimes the politics or organization of a company's business lends itself to a federated portal solution. Different groups can administer and control their own content separately using smaller systems that require less planning and maintenance. This can also be accomplished in a large portal system to some degree by having different departments control remote servers that serve secure content to the portal. In a federated portal system, information is shared via federated search and, possibly, shared portlets. For these systems, identify the scaling needs of each portal in the network and decide how the portals should be connected. It is very important that the various groups agree on how content is to be shared and how much load they can expect other portals to place on their portal. Size each system as you would a single large portal, but take into account potentially higher load on shared remote servers and federated search pages.

## High Availability Deployment

### Portal Server Load Balancing

The portal can be used with any load balancing system that supports sticky IPs, such as Cisco LocalDirector, F5 Big-IP, and Windows NLB load balancing systems. Session states are maintained on the Plumtree Web servers themselves. Therefore, if a Web server is taken out of the Web farm, sessions on that server are lost. If users have not set their Web browser to Remember My Password, they will have to log back in to the portal.

It is possible for the portal to become unresponsive while the Web site is still operational. In that case, the load balancer should assume that the portal is still operational and continue to send requests. The load balancer should perform content verification to ensure that the portal is actually available.

Since users use the Portal Servers in different ways, the load balancer should send requests to the computer with the most available resources instead of simply performing a round-robin distribution of requests.

For maximum fault tolerance, Plumtree recommends that load balancers be clustered, so if one load balancer fails, another will continue to distribute requests. Consult manufacturer guidelines on clustering load balancers.

## Database Server Load Balancing

The database server can be scaled using any database-compatible clustering technology. Currently, this means that scaling can only be provided by a larger machine. If necessary, each portal database can be placed on a separate computer and scaled separately. If running on Windows, failover of databases can be provided with Microsoft Cluster Services, and geographic load balancing and failover can be provided using SQL Server replication. However, this method is technically and administratively challenging and is not recommended unless availability requirements cannot be met otherwise.

Oracle databases can be deployed for high availability. Plumtree does not support either client-side failover or Oracle RAC.

## Search Server Load Balancing

The Search Server can be scaled using two methods, which can be used in combination:

- Read/write split load balancing

One instance of Search Server is designated the “index” or “write” server. The other is designated the “query” or “read” server. Updates are directed to the index server, while queries are directed to the query server. New data on the index server is periodically replicated to the query server. Third-party file transfer and job scheduling technology must be added to the standard Search Server to enable the replication. (You must use the Search Server’s Replicate utility to copy a consistent set of search indices; you cannot use regular Windows file system copy without risking index corruption. For more information on the Search Server’s Replicate utility, refer to Plumtree Knowledge



Base article 11875: “Replicating a 5.0 Search Collection”). Job scheduling can be handled by the portal system if using version 5.0.

- Read/read split load balancing

“Query” or “read” servers can be balanced using an IP load-balancing technology like NLB or a hardware load balancer. The IP load balancer will also implicitly provide failover for the “read” server.

- Read failover

A server can be designated to the portal as the “query” or “read” failover server. No additional load balancing or failover technology is necessary to implement this.

- Write load balancing

There is no way to provide load balancing for the “write” server other than increasing the machine size. If there is concern about write performance, query load should be completely removed from the machine using read/write split load balancing.

## Automation Server Load Balancing

Automation Servers do not require any special technology to provide load balancing or failover. Installing multiple instances of the Automation Servers in a portal system will provide load balancing, as jobs can be designated to run on any set of available servers. In case of server failure mid-job, the job will not complete on another server. However, jobs are typically scheduled to recur, and the next instance of any standard Plumtree job will complete the processing.

Automation Servers can be load balanced by registering job folders to multiple Automation Servers. The Automation Servers poll the database and pick up the next available job. Should one Automation Server fail, another Automation Server will run the necessary jobs.

## Remote Server Load Balancing

Remote servers can be load balanced using Parallel Portal Engine load balancing. Refer to [“Parallel Portal Engine \(PPE\)” on page 3-67](#) for instructions on configuring this feature. Remote servers can also be load balanced in a similar way to Portal Servers using the same kind of load balancing hardware.

## Collaboration Server Load Balancing

Collaboration Server 3.0 supports clustering to provide failover and load balancing. In clustering mode, multiple Collaboration Servers will communicate with each other to maintain a single, consistent, logical image. To enable clustering, you must configure the portal and then modify Collaboration Server configuration.

The portal provides load balancing through mapping one domain name to multiple IP addresses. You must provide a single domain name that contains the IP addresses of each Collaboration Server that you want to participate in the cluster. This is the name that will be used as the portlet remote server name.

The Collaboration Server settings for clustering are located in two files in the **config** directory: **config.xml** and **cluster.xml**. Edit the config.xml file and change the following:

1. Enable clustering by changing the following line:

```
<cluster enabled="no">cluster.xml</cluster>
```

to:

```
<cluster enabled="yes">cluster.xml</cluster>
```

2. Save the file and restart the server to apply the changes. This step must be done on each Collaboration Server instance that will participate in the cluster.

By default, Collaboration Server uses UDP multicasting for communicating between servers. This option is sufficient for most deployments and is the most efficient. In environments where UDP multicasting is not allowed, you can configure Collaboration Server to use UDP unicasting using the following steps:

1. Edit **cluster.xml**. You need to nominate one of the machines in the cluster to be the coordinator:

```
<coordinator-host>machine.name.goes.here</coordinator-host>
<coordinator-port>9990</coordinator-port>
```

The port number can be any free port number.

2. Change the cluster profile to lan-cluster

```
<profiles profile='lan-multicast-cluster'>
```

to

```
<profiles profile='lan-cluster'>
```

3. Save the file and restart Collaboration Server to apply the settings. This step must be done on each Collaboration Server instance that will participate in the cluster.

## Content Server Load Balancing

Content Server cannot be load balanced and does not support high availability configurations using clustering or database replication.

## Studio Server Load Balancing

Studio Server does not currently support any form of load balancing or clustering. That said, it is possible to configure multiple Studio Servers to handle failover and scalability issues.

Configuring Studio Server for failover involves setting up a “hot-standby” computer that can be switched over to if the primary computer hosting Studio Server is no longer able to handle requests. How to determine when the primary server has failed, and that the hot-standby should take over is beyond the scope of this document, however.

Multiple computers hosting Studio Server can also be configured to provide additional scalability. Unlike a typical clustering setup, where multiple computers handle requests for the same application data objects in parallel, configuring Studio Server for scalability involves segmenting portlets across separate computers. For example, in a single-computer setup, a given computer with Studio Server might be hosting hundreds of user-created portlets. An administrator could set up a second computer hosting Studio Server (using a separate database from the first Studio Server). The administrator could then designate this computer as the host for any additional Studio Server portlets created on the portal.

## Analytics Server Load Balancing

Analytics Server does not currently support any form of load balancing or clustering.

## Document Repository Load Balancing

Multiple instances of the Document Repository can be load balanced and failed over using IP load balancing such as NLB or a hardware load balancer. This will also provide partial failover for the Document Repository. However, the Document Repository computer requires a single writable file system backing store. This backing store cannot be load balanced, but it can be failed over with one of the following:

- A shared local disk, failed over via MSCS
- An external shared network drive, implemented using either NAS or MSCS

## External Service Load Balancing

The portal will be dependent on many other servers and services to function. Each of these services must provide some failover. At a minimum, these services will include:

- Authentication sources such as the customer/partner directory and the employee domain
- File and Web servers housing documents and other content
- External applications to which portlets provide access

## Parallel Portal Engine (PPE)

The PPE Load Balancer (PPE-LB) is a built in feature that allows you to load balance your Remote Servers to make better use of the Parallel Portal Engine (PPE). PPE-LB is Plumtree's load balancing solution for middle-tier HTTP messaging (between the Portal Server and the Remote Servers). It provides robust failover services for high availability and eliminates the need for a third party load balancing solution in front of Portlets. PPE-LB is designed to be as easy to configure as round robin DNS and readily solves proxy and SSL problems that are typically encountered with load balancing devices in middle-tier messaging.

There are two methods for configuring PPE load balancing:

- on the DNS server. (Refer to [“Configuring PPE Load Balancing on the DNS Server” on page 3-67.](#))
- if you are running on Windows, through the registry. (Refer to [“Configuring PPE Load Balancing Through the Registry” on page 3-68.](#))

### *Configuring PPE Load Balancing on the DNS Server*

On the DNS server, configure the Remote Server cluster name (for example, gs.portal.company.com) to resolve to multiple IP addresses. This is similar to setting up DNS round robin, except that PPE load balancing will failover, provide stickiness, and act as a

load balancer. Each Remote Server in a cluster must have a unique IP address and must have the same software installed.



**Note:** Editing the hosts file on a Windows NT machine is not equivalent to configuring the DNS server. Windows NT caches and returns only the first IP address, instead of returning multiple IP addresses the way a DNS server does. If you are not able to configure the DNS server, contact Customer Support for registry settings you can add to provide equivalent functionality.

The entry in the DNS server should look something like this using BIND on a Unix DNS server:

```
remoteserver      60      IN      A       10.10.10.1
remoteserver      60      IN      A       10.10.10.2
remoteserver      60      IN      A       10.10.10.3
```

If the domain is *company.com*, then the *remoteserver.company.com* host name should be resolved to this list of IP addresses by the DNS server.

### *Configuring PPE Load Balancing Through the Registry*

1. In the registry, create the following key (using a program such as **regedt32**):
 

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
  Services\Plumtree\HostMappings
```
2. Add a value to this key.
3. Set the data type to REG\_MULTI\_SZ.
4. Set the value to the main URL to which requests will be directed (for example, *remoteserver.plumtree.com*). You will need to create a Remote Server pointing to this URL.
5. Click **OK**.
6. Set the multiple values for this key to the Remote Servers you want to load balance; separate values with line breaks.

For example, to load balance `remoteserver.plumtree.com` by alternating requests to `GS1.plumtree.com` and `GS2.plumtree.com`, type `GS1.plumtree.com`, press ENTER, type `GS2.plumtree.com`, and then click **OK**.

### *Portlet Support*

Most Portlets should work correctly with PPE load balancing, but some Portlets may do in-memory caching that assumes the underlying database will not be modified by another application. Consult the Portlet documentation or Portlet developer to determine if specific Portlets can be load balanced.

### *PPE Load Balancing and SSL*

If your Remote Servers use Secure Sockets Layer (SSL), Plumtree recommends creating a single SSL certificate by name and adding it to each machine in a Remote Server cluster.

### *Verifying That PPE Load Balancing is Configured Correctly*

You can verify the DNS server configuration by using a tool called *nslookup*. For example, try using *nslookup* on `www.microsoft.com`:

1. Open a command line prompt.
2. Run *nslookup*. At the command prompt enter:

```
nslookup www.microsoft.com
```

This command will return something similar to the following lines:

```
Server:  plumdcl.plumtree.com
Address:  10.1.88.4

Non-authoritative answer:
Name:     www.microsoft.akadns.net
Addresses: 207.46.197.100, 207.46.197.102, 207.46.230.218
Aliases:  www.microsoft.com
```

Notice that `www.microsoft.com` is using round robin DNS and three different IP addresses.

The PPE updates itself from the DNS server. The PPE algorithm refreshes the list of IP addresses in a Remote Server cluster more frequently as more load is placed on it; it is not based on a timed update. It starts load balancing without requiring you to restart the server.

Testing against a Guest user My Page will not trigger load balancing (since it is sticky on a per-session basis). You will need to test with unique users' My Pages under load. If you do not have tools to do load testing on a My Page with unique users, you can use Homer. There are ASP pages on the CD-ROM under **\Unsupported\Plumtree\WebTesting** that help with this.

## *PPE Load Balancing: Frequently Asked Questions*

### General Questions

#### 1. What is PPE Load Balancing (PPE-LB)?

PPE-LB is Plumtree's load balancing solution for middle-tier HTTP messaging (between the Portal Server and the Remote Servers). It provides robust failover services for high availability and eliminates the need for a third party load balancing solution in front of Portlets. PPE-LB is designed to be as easy to configure as round robin DNS and readily solves proxy and SSL problems that are typically encountered with load balancing devices in middle-tier messaging.

#### 2. Can I use my own load balancer (such as, F5 Big-IP or Cisco LocalDirector) for Portlets?

Yes, absolutely. Plumtree is providing a robust, low-hassle solution that will fit most portal customers' needs. If, however, a hardware load balancing device such as F5 Big-IP or Cisco LocalDirector is preferred, there are no Plumtree issues. It will simply be more expensive to purchase, configure, and maintain.

In some situations, a load balancing networking device may be necessary. For example, if a Remote Server were exposed on Internet for access by partners, it would make sense that the owners of the Remote Server would want control over the load balancing device. Since PPE-LB logic and failover is not performed on the Remote Servers but



on the Portal Servers, it is not a good solution for the owners of the Portlet in this case.

3. Why would I use PPE Load Balancing instead of another load balancing device for Portlets?

PPE-LB is built into the Plumtree PPE messaging engine. This allows our customers to avoid buying, configuring, and maintaining a separate load balancing device. Going forward, PPE-LB will continue to be dedicated to solving the load balancing and high availability issues with middle-tier HTTP messaging, especially as Web services become more and more prominent. This focus will continue to provide the best integrated solutions and the lowest total cost of ownership.

Technically, PPE-LB performs its load balancing and failover algorithms on the Portal Web Server. This eliminates the need to have software configured on the Remote Servers and eliminates the need for a networked load balancing device in the middle. It also is integrated with the PPE request engine and can utilize more portal user information to make smarter, more efficient requests. For example, the proxy problem is easily solved since the PPE knows which user is making the request and sticky clients is readily and easily supported in the load balancing, unlike networking devices which must sniff and re-route the HTTP requests.

In summary, the table of Portlet Server load balancing options looks something like this:

Table 3-12: Load balancing options

	PPE-LB	WLBS/NLB	Other (such as, F5, Cisco)
Sticky Portlet	Yes	No	Maybe
Non-sticky Portlet	Yes	Yes	Yes
Sticky SSL Portlet	Yes	No	Maybe
Non-sticky SSL Portlet	Yes	Yes	Yes

The *Maybes* depend on the load balancer and may require quite a bit of configuration. For example, F5's Big-IP (<http://www.f5.com/solutions/techbriefs/index.html>) provides a way to have the load balancer insert a cookie into the HTTP headers to provide both

stickiness and SSL support. Recent versions of Cisco LocalDirector can do this too ([http://www.cisco.com/warp/public/cc/pd/cxsr/400/prodlit/1215\\_pp.htm](http://www.cisco.com/warp/public/cc/pd/cxsr/400/prodlit/1215_pp.htm)). Plumtree has had difficulties at times trying to make LocalDirector do this properly. Microsoft WLBS/NLB only switches off of incoming IP address, so it does not load balance sticky Portlets properly. Microsoft has a product called Application Center 2000 that has a “Request Forwarder” ISAPI filter that is supposed to solve this.

4. What features of third party load balancers are not supported by PPE-LB?

PPE-LB is not intended to replace third party load balancers generally. Load balancers are still commonly used to balance the load across a farm of Portal Servers.

Architecturally, load balancers traditionally must behave like a networking device (a switch or router) and redirect traffic as needed. As such, features such as URL Parsing, Layer 7 Switching, Cookie Persistence, and basic IP switching do not make any sense in a middle-tier load balancing product like PPE-LB; they only make sense in a networking device.

However, some features are applicable to the middle-tier load balancing problem. PPE-LB currently does not support a defined way to configure weights on different servers. It assumes that all Portlet Servers in a cluster have the same performance characteristic and weights them equally. Also, many load balancing devices have dynamic tools to do rolling updates where servers are taken offline for update incrementally.

5. What is the proxy problem?

This refers to the problem where incoming requests from users using proxy servers will all come from the same IP address: the proxy's. Most load balancing devices that support sticky clients use the incoming IP address to send requests to the different nodes in a Web farm. Since all the requests are coming from a proxy server, this means that all requests are sent to the same node- not good. Microsoft WLBS/NLB is an example of a load balancer that cannot handle the proxy problem. Other load balancing devices such as F5 Big-IP add cookies to the HTTP headers that allow it to perform sticky clients without relying on the incoming IP address. Others such as Cisco LocalDirector can parse the HTTP header and use portions of it to hash instead of the incoming IP address. For example, in Plumtree, the UserID argument might be used.

The proxy problem is sometimes encountered with the Portal Server, but with Portlet Servers, it is encountered every time. All Portlets are sent requests from the Portal Server, so any sticky client load balancing device that relies on incoming IP addresses to do this will fail.

6. Why does SSL cause problems for some load balancers?

SSL encrypts the headers. Load balancing devices that act as networking devices will read the packets and perform their load balancing algorithms based on the content. If SSL is used and the packets cannot be read, the load balancing device will not work. Often, the load balancing device itself must be configured to implement SSL so it can decrypt the HTTP request and perform its load balancing. Most load balancing devices can do this, but it requires much more advanced knowledge of the device and often causes unexpected difficulties.

### Technical Questions

1. What performance overhead is there when enabling PPE-LB? Do I need bigger Portal Web Server boxes?

There is no performance impact. Extra processing power is not necessary on the Portal Servers.

2. Why is PPE-LB using DNS for configuration?

Configuring DNS is easy and familiar to administrators. Most customers can readily configure DNS to do round robin DNS. PPE-LB uses the DNS one-to-many mapping of host name to multiple IP addresses to define a Portlet Server cluster.

As an alternative to DNS, registry settings can be set on each PWS to define the list of IP addresses mapping to a host name. The registry key is **HKLM\SYSTEM\Current-ControlSet\Services\Plumtree\HostMappings**. Using **regedt32**, multiple **REG\_MULTI\_SZ** values should be created on this key. The name of the value should be the host name (for example, gs.company.com) and the value should be a list of IP addresses. Note that in **regedt32**, the **REG\_MULTI\_SZ** type is shown in a multi-line text box and each entry should be on a separate line for proper delimiting.

### 3. Isn't PPE Load Balancing just round robin DNS then?

No. Round robin DNS relies on Web browser clients asking their DNS server for the IP address of the server address and rotating users through a list of IP addresses. PPE-LB asks the DNS server for all IP addresses that resolve to a given address (similar to an `nslookup` call) and stores that list in memory. It then performs load balancing and failover as necessary. No further DNS lookups are made and, unlike round robin DNS, failure detection and failover is performed.

### 4. What is PPE-LB's balancing algorithm?

PPE-LB uses the combination of host name and port number (for example, in `https://gs.company.com:8080/xyz/portlet.asp`, the `gs.company.com` and `8080` values) for stickiness on a per-user basis. In Plumtree, stickiness is always enabled. If multiple Portlets are placed on the same Portlet Server cluster and the Portlet URL uses the same host name and port number, all requests should stick to that machine. This may mean that a My Page with five Portlets, all with the same host name and port number in the URL, will have all five requests go to the same machine. Click-throughs using the Portlet gateway (preferences pages) should stick to the same machine.

Internally, PPE-LB selects the node to stick a user to when a request to a particular host name and port number is first made. This first request can be from a Remote Portlet on a My Page, Community page or from a Federated Search request. The node is chosen with a least-used algorithm and is randomly selected if all servers are at the same usage levels. *Used* here includes usage resulting from each request sent to a particular IP address in the cluster; it is not just a counter of user-to-IP-address connections.

### 5. What triggers a failover?

Network errors (such as, inability to connect, read timeouts, PING (ICMP) host unreachable messages); basically, low-level TCP/IP and network errors.

HTTP timeouts might or might not be considered a failover condition. If connection is successful but no data is returned within 75% of the timeout length, it is considered a failover, and an attempt to switch to an alternate node is made. To guarantee no failovers in cases where Portlets take a long time to execute and might have frequent

timeouts, the HTTP response header should not be buffered and should be returned immediately.

Many times, a Web server will fail because of bugs in the Web server scripting. These bugs may cause memory corruption, server hangs, and other errors that can prevent the PPE-LB from knowing that the server node has failed. A customer should consider using a third-party monitoring tool. Monitoring tools can be configured to identify faulty situations and deal with the problem by restarting the Web server or even the machine (for example, if *ASP: Requests Executing* performance monitor counters reach 25 and *ASP: Requests Queued* reaches 100, restart the IIS service). These are things that PPE-LB cannot do, so the third-party tool is recommended.

6. How fast is the failover from one node to the next?

PPE-LB always attempts to switch over immediately. It attempts to detect the time out or error and failover any current requests to the next node. For end-users, this usually means that there is no interruption in service.

7. If a failed server is restarted, when does the PPE-LB begin using it again?

If a server fails and PPE-LB stops sending requests to the failed server, the restarted server will be incorporated back into the cluster when a significant number of new users log in. Old users will stick to their previous Portlet Server and new users will trigger PPE-LB to update its internal list of available nodes in the cluster. Because this update happens on-demand, it may be very quick or it may take some time before requests are sent to the restarted node.

8. With multiple Portal Servers all running the PPE-LB engine, how is the load balanced?

There is no communication between Portal Servers, where the load balancing logic is performed. The result is that there can be spikes where load is concentrated on a particular Portlet Server. However, in practice, nominal load on a Portlet Server should be in the 20-40% range and spikes from requests are common and readily handled by Web servers. So the load is as evenly balanced across Portlet Servers as any normal Web load.

## 9. What happens to state information for a user on a Portlet Server failover?

The Portal Server in-memory state is retained but may not make sense to the Portlet Server that the user rolls over to. A Portlet Server may have returned non-persistent information such as ASPSESSIONID or JSESSIONID cookies that will not refer to a valid ASP or JSP session on the second Portlet Server machine.

The Portlet developer has the option of building Portlets to avoid loss of in-memory state (for example, using Application Server high-availability features) or to be able to re-create in-memory state in the case of a failure. For many applications, a stateless Portlet Server is not possible. For example, creating a Siebel session in a Siebel Portlet may be unavoidable, and the session connection probably cannot be easily replicated across servers. But re-creating a Siebel session on demand using the preferences information passed from Portal Server to Portlet Server is a good idea. On the first access by a user or when a failover to a second Portlet Server machine is done by PPE-LB, the end-user does not have an interruption in service and state can be saved using files, database, or application server techniques.

Portlet developers should refer to the latest Enterprise Web Development Guide for tips and best practices.

## 10. What is the best way to test PPE-LB in a customer deployment?

There are several common practices to verify the load balancing is working:

- a. Do an nslookup on the Portlet Server cluster host name to verify that the cluster is defined properly. This should return a list of multiple IP addresses.
- b. Create a test Portlet that points to a static HTM page on every machine with the machine's name on it. Perform the following manual test with a Web browser: login, hit My Page several times, and then logout. Perform the test again to confirm that the PPE-LB switches between different Portlet Servers.
- c. After step b., unplug the network connection to a Portlet Server node that you connected to. Hit refresh and see if your connection is failed over to another node in the cluster.

In general, heavy load testing is not recommended because it is difficult to setup the test properly. If a load test is required, make sure to monitor the number of hits on

each Web server and verify the load is roughly well distributed. The load test script should be sure to simulate the login, My Page hits, logout sequence. If the same session is used over and over again (as in a Guest user My Page load test) then the stickiness will be in effect and all load will be weighted to one Portlet Server. Proper test script creation and proper monitoring of the Portlet Servers is necessary. Double-check everything and be very detailed in understanding what should happen prior to each test run.

#### 11. What issues should Portlet developers be aware of?

Knowing how session state is and can be maintained is very important. It is important to identify *transient* session state versus *transactional* session state and the need for stickiness and transactional integrity. PPE-LB automatically supports sticky clients, so there is no issue with using in-memory session state to cache information. However, if a Portlet is holding transactional information such as a shopping cart, then it should take advantage of underlying application servers and databases to persist such information in case the Portlet Server fails and the user is rolled over to another machine. Refer to the *Enterprise Web Development Guide* for more details on Portlet development issues in a load balanced environment.

#### Administrative Questions

##### 1. Do I still need to do monitoring of the Portlets?

Yes, failures in Portlet Servers are more often from bugs in the code than hardware. So monitoring the status of the servers is essential to any high-availability strategy.

##### 2. How do I add machines to the cluster at peak load times and remove them at others, or take machines out to upgrade them with new Portlet fixes?

Stopping the Web server will cause the PPE-LB to failover existing connections to another machine in the cluster. Upgrading a box involves stopping the Web server, upgrading the components, and then re-starting the Web server. PPE-LB will begin sending requests to the machine after a short time. This is known as doing a rolling update. Note that one Portlet Server can be slightly different than another during the update window and the users will not notice since they will stick to their machine. Also, refer to question 7 on page 3-76.



3. Is there anything in the Plumtree Portlet object that needs to be configured?

No. The Portlet object stores the URL of the Portlet Server. The host name and port number combination is the only information PPE-LB needs. Note that preferences pages, Portlet pages, and gatewayed pages should all stick to the same node in a Portlet Server cluster using PPE-LB.

4. Do I need to maintain mirrored copies of Portlet code on my Remote Servers?

Yes. As in any Web farm configuration, it is expected that the nodes be functionally identical so that all users see the same Web service.

5. Are Portlets certified by Plumtree for use with PPE Load Balancing?

No. Portlet developers should document load balancing configuration support with each Portlet. Plumtree may add additional options in the future for Portlet certification with load balanced configurations.

Note that using PPE-LB to load balance Portlet Servers is no different than using a Cisco or F5 product as far as the Portlet developer is concerned. The Portlet developer should be concerned with designing around sticky/non-sticky sessions and, in the sticky case, dealing with failovers. For complex Portlets with state, this can be a complex issue that would require good documentation so that testing can be qualified by the types of high availability features supported by the Portlets.

6. Do I need to configure the Portlet Servers to use the same SSL certificate?

If the SSL certificate installed on the Portlet Server Web servers is *by name*, then it can be the same or it can be different on each Portlet Server machine. If the certificate is issued *by IP address*, then they should be different on each Portlet Server machine.

7. What is the maximum number of Portlet Servers that PPE-LB can work with?

There is not limit in the PPE-LB algorithm or DNS configuration, but Plumtree recommends that customers employ 32 or less Portlet Server machines in a cluster.

### *PPE Configuration Settings for Plumtree Corporate Portal 5.x for Windows*

All Plumtree Parallel Engine (PPE) settings are located in the registry under the "HKLM\SOFTWARE\Plumtree\3.0\HTTPLib" key.

If present, the following values allow fine-tuning the PPE:

- **ForceConnectionClose** (REG\_DWORD, default value: 0) - If set to 1, the socket will be closed after sending a single HTTP request, effectively making PPE use HTTP 1.0 (no connection pooling).
- **ForceAuthenticatedConnectionClose** (REG\_DWORD, default value: 0) - If set to 1, the socket will be closed after sending a single HTTP request on an SSL connection, effectively making PPE use HTTP 1.0 for SSL (no connection pooling).
- **RequireServerCertificateForSSL** (REG\_DWORD, default value: 0) - If set to 1, the PPE will require a server certificate to establish an SSL connection. If set to 0, the PPE can communicate to SSL servers providing only message integrity and privacy but not authentication.
- **DNSCacheDefaultTTL** (REG\_DWORD, default value: 300000) - TTL (in milliseconds) for DNS cache entries. If this amount of time passes after resolving a name to an address, a new call is made to the operating system to obtain up-to-date IP addresses for the given name.
- **HostCacheDefaultTTL** (REG\_DWORD, default value: 1800000) - Frequency (in milliseconds) at which the socket cache will be refreshed. This time defines "stale sockets" garbage collection period.
- **ErrorThreadStall** (REG\_DWORD, default value: 10) - Delay (in milliseconds) before reporting network errors to the API user. Allows OS to clear network buffers before retry is made.
- **ValidateServerCertificates** (REG\_DWORD, default value: 1) - If set to 0, SSL certificates will not be validated. If set to 1 certificates will be checked almost completely (no revocation check performed for performance reasons).
- **ForceMutualAuthentication** (REG\_DWORD, default value: 0) - If set to 1, SSL will initiate full authentication (including client) during connection establishment even if the server does not require it.

- **UseClientCertificates** (REG\_DWORD, default value: 0) - If set to 1, SSL sends default certificate for the current NT user to the server.
- **UseFastDNS** (REG\_DWORD, default value: 0) - If set to 1, only single DNS lookup will be performed (vs. DNS+WINS+...). This can substantially speed up name resolution (and eliminate "first try - can't resolve" problem) on a network with a good DNS server.
- **GetAndLockSockStall** (REG\_DWORD, default value: 10) - Socket reconnect period (in milliseconds). If set too short, and if errors are present on the network, high (and unnecessary) CPU usage is possible.
- **BinGWThreshold** (REG\_DWORD, default value: 16386) - Minimum data length (in bytes) to be considered for routing through binary gateway. If set to 0xFFFFFFFF gateway logic is completely disabled in WebDownloaderParallel, and ISAPI filter can be safely removed from the portal.
- **MaxPostData** (REG\_DWORD, default value: 52428800) - Maximum data length (in bytes) to be accepted by Plumtree gateway for processing. Note: This limit applies to both binary and non-binary (ASP) uploads. If data length exceeds this value, the user will be redirected to an error page.
- **BinGWTimeout** (REG\_DWORD, default value: 2400000) - Maximum time (in milliseconds) for gateway to process a single request. Note: We recommend that you not increase this value even for modem-connected clients. Too many lingering requests might prevent IIS from accepting more connections from browsers.
- **BinGWPacketTimeout** (REG\_DWORD, default value: 10000) - Maximum time (in milliseconds) for gateway to wait for next chunk of response body to arrive from the Remote Server (referred to as Gadget Server in 4.5WS). Used to detect hanging servers and network outages.
- **BinGWTempDir** (REG\_SZ, default value: default system temporary directory) - Location for temporary files for upload.

## Security Strategies

Providing portal services to external users in a secure fashion is a very common requirement. However, there is no perfect solution. Exposing internal resources to the outside world almost always introduces security vulnerabilities, regardless of the application being deployed. The fundamental question is how much and in what ways are you willing to compromise (and how much you are willing to spend) to provide this functionality. There are a number of ways to implement an externally facing portal. This section contains a discussion of some of the issues and Plumtree functionality pertinent to those issues. Plumtree does not endorse any single strategy. Your company must make the final decisions around policies, procedures, budget, and hardware.

### Configuring SSL for Your Enterprise Web Deployment

Securing your portal is no easy task. This section describes how to set up your Enterprise Web deployment to run in security modes 1, 2, or 3. (Security mode 0 is the default mode.)

## Portal Security Modes

The following security modes are available:

- **Security Mode 0:** Portal pages remain in whatever security mode, http or https, that the user initially accesses the portal. For example, if a user accesses the portal via http, all the portal pages will remain http; if a user accesses the portal via https, all the portal pages will remain https. This is the default setting.

Use this mode only when the deployment is used internally, behind a firewall, and without an SSL accelerator. For example, you might want to use this mode for testing or development deployments.

- **Security Mode 1:** Certain portal pages are always secured via SSL and other pages are not. For example, the login page may always be secured but a directory browsing page may not. The page types that are secured are configurable.

This mode is not generally recommended.

- **Security Mode 2:** All portal pages are always secured via SSL, that is, pages are accessed via https.

Use this mode if you are not using an SSL accelerator. In this mode, the Web server should provide an SSL endpoint. We recommend against configuring the SSL endpoint directly on the Tomcat application server. Although application servers can handle Web requests directly, for scalability and security reasons, we recommended that you not permit your users to connect to the application server directly. Instead of securing the application server itself, you secure the front-end Web server and the channel between the Web server and the application server. Therefore, you must set up SSL and install an SSL certificate on the Web server.

- **Security Mode 3:** The portal uses an SSL accelerator.

This is the most common set up for production deployments. Use this mode if you are using an SSL accelerator. As with Security Mode 2, users are not connecting to the application server directly, so you need to secure the front-end application server and the channel between the accelerator and the application sever. Therefore, you must set up SSL and install an SSL certificate on the SSL accelerator.

## Roadmap to Changing Security Modes and Setting Up SSL

There are several steps involved in setting up SSL for your Enterprise Web deployment. This section provides a brief overview of the steps you need to complete.

1. Set up SSL on the Web servers or SSL accelerators that run the Portal Server and Image Server. Refer to your Web server or application server documentation for instructions on setting up SSL and creating, signing, and installing an SSL certificate.
2. Configure the Portal Server by editing the configuration file—`j_config.xml` for Java deployments, `n_config.xml` for .NET deployments. The configuration file is located in your portal installation directory, for example, `C:\Program Files\plumtree\ptportal\5.0\settings\config\j_config.xml`.
  - a. Make sure `HTTPSecurePort` and `HTTPPort` are set to the ports you want to use.
  - b. Change `ApplicationURL0` from `*` to  
`http://computer_name:port_number/portal/server.pt.`




**Note:** You do not need to include the `port_number` for .NET deployments.

- c. Change `SecureApplicationURL0` from `*` to  
`https://computer_name:port_number/portal/server.pt.`



**Note:** You do not need to include the `port_number` for .NET deployments.

- d. If you have more than one URL mapping entry, you might need to change those entries as well. Refer to the comments in the configuration file for more information on URL mapping.
  - e. Change `SecurityMode` from `0` to `1`, `2`, or `3`.
  - f. Change `ImageServerSecureBaseURL` from `http` to `https`, and change the Image Server port to the correct one.

3. If you set the `IMAGESERVERCONNECTIONURL` in the portal configuration file to an Image Server running in SSL (not recommended) you must import onto the Portal Server the certificate of the CA that signed the certificate used by the Image Server:
    - For Java portals, you need to use Java's `keytool` utility to import into Java's cacerts store. Refer to [“Importing CA Certificates into the cacerts Keystore \(for Java Portals\)” on page 3-86.](#)
    - For .NET portals, you need to import the CA certificate into MMC. Refer to [“Importing CA Certificates into MMC \(for .NET Portals\)” on page 3-87.](#)
-  **Note:** If you have any portlets or Remote Servers that use JSControls or Adaptive Portlets you must also import the CA certificate into those runtimes. (The JSControls libraries are embedded in server and EDK products and are initialized by HTTP-downloading an XML config file from the Image Server.) Refer to the sections on page 3-86 or page 3-87 (referred to in this step).
4. An MPPE SSL issue unrelated to portal security modes, if you have a Remote Server (a server running remote Web Services such as portlets, authentication sources, profile sources, or crawlers) running in SSL, you need to import onto the Portal Server the certificate of the CA that signed the certificate used by the Remote Server. Refer to the sections on page 3-86 or page 3-87 (referred to in Step 3).
  5. If you are running Content Server, Workflow Server, Collaboration Server, or Studio Server, refer to the following sections to configure them to use a secure Image Server and Portal Server:
    - [“Setting Up Content Server to Use a Secure Image Server or Portal Server” on page 3-88](#)
    - [“Setting Up Workflow Server to Use a Secure Image Server or Portal Server” on page 3-89](#)
    - [“Setting Up Collaboration Server to Use a Secure Image Server” on page 3-90](#)
    - [“Setting Up Studio Server to Use a Secure Image Server or Portal Server” on page 3-90](#)

## Importing CA Certificates into the cacerts Keystore (for Java Portals)

For each machine that makes requests to a secured server running in SSL, you must import into the cacerts keystore the certificate of the CA that signed the certificate used by the secured server:

1. On the machine that makes requests to a secured server, open a command prompt.
2. Copy the CA certificate to this machine.

To obtain the CA certificate, navigate to the CA and save the .der encoded certificate file as a .cer file; you might want to use *imgsvr.cer* for an Image Server or *portal.cer* for a Portal Server, or you might want to use the server hostname.

3. Import the certificate:

```
keytool -v -import -trustcacerts -alias CA_alias -file  
CA_certificate_path -keystore cacerts_keystore_path
```

Replace the variables with the following information:

- *CA\_alias* - the alias for your CA, for example, *verisign* or the server hostname
- *CA\_certificate\_path* - the path and filename to the CA certificate you copied to the Portal Server
- *cacerts\_keystore\_path* - the path to your cacerts keystore, located at "jre\lib\security\cacerts" in the home of the JVM that runs your Java application server, for example:
  - For Tomcat, jakarta-tomcat-4.1.30-LE-j2sdk1.4.1\_05-win32\j2sdk1.4.1\_05\jre\lib\security\cacerts
  - For WebLogic, bea\weblogic700\server\lib\cacerts
  - For WebSphere, java\jre\lib\security\cacerts

4. Enter the password to the cacerts keystore.



## Importing CA Certificates into MMC (for .NET Portals)

For each machine that makes requests to a secured server running in SSL, you must import into the MMC the certificate of the CA that signed the certificate used by the secured server:

1. On the machine that makes requests to a secured server, open a command prompt.
2. Copy the CA certificate to this machine.

To obtain the CA certificate, navigate to the CA and save the .der encoded certificate file as a .cer file; you might want to use *imgsvr.cer* for an Image Server or *portal.cer* for a Portal Server, or you might want to use the server hostname.

3. Open MMC:

```
C : \>mmc
```

4. Click **Console | Add/Remove Snap-in**.
5. Click **Add**.
6. Click **Certificates**.
7. Click **Computer Account** and then click **Next**.
8. Click **local computer** and then click **Finish**.
9. Click **Close** to close the Add Standalone Snap-in dialog box.
10. Click **OK** to close the Add/Remove Snap-in dialog box.
11. In the MMC tree, expand to **Console Root | Certificates | Trusted Root Certificate Authorities | Certificates**.
12. Right-click **Certificates** and select **All Tasks | Import**.
13. Click **Next**.
14. Select your certificate.
15. Click **Next**.
16. Choose to place all certificates in the following store: **Trusted Root Certification Authorities**.
17. Click **Next** and then click **Finish**.
18. Restart IIS.

## Setting Up Content Server to Use a Secure Image Server or Portal Server

- I. If you are using a secure Image Server:
  - a. In a text editor, open **content.properties** (located in your Content Server installation directory, for example, C:\Program Files\plumtree\ptcs\6.0\settings\config\content.properties).
  - b. Change Image Server entries:
    - If you are using Security Modes 1 or 2, find and replace *all* occurrences of `http://machine_name/imageserver` with `https://machine_name/imageserver`, where *machine\_name* is the name of the machine hosting Content Server.
    - If you are using Security Mode 3, change the Image Server entries as follows (note that some variables use http and some use https):
      - `CommunityImagePublishBrowserLocation=https://machine_name/imageserver/plumtree/portal/templates`
      - `CommunityImagePreviewBrowserLocation=https://machine_name/imageserver/plumtree/portal/templates/preview`
      - `CommunityStyleSheetListURL=http://machine_name/imageserver/plumtree/common/public/css/community-themes.txt`
      - `JSComponents.AlternateImageUrl=http://machine_name/imageserver`
- If the Image Server is on Java, be sure to change the port to the correct one (for example, `https://machine_name:ssl_port_number/imageserver`).
2. If you are using Security Modes 1 or 2, import the certificate of the CA that signed the Image Server and/or Portal Server certificate into Content Server:
  - For Java portals, refer to [“Importing CA Certificates into the cacerts Keystore \(for Java Portals\)” on page 3-86](#).
  - For .NET portals, refer to [“Importing CA Certificates into MMC \(for .NET Portals\)” on page 3-87](#).

3. If Content Server runs on WebSphere against a .NET portal in Security Mode I, complete the following steps:
  - a. Open the WebSphere Admin console.
  - b. Navigate to the Default Server.
  - c. Click the **JVM Settings** tab.
  - d. Under System Properties, click **Add** (this should add a line).
  - e. For **Name** type “java.protocol.handler.pkgs” and for **Value** type “com.ibm.net.ssl.www.protocol”.
4. Restart Content Server.

### Setting Up Workflow Server to Use a Secure Image Server or Portal Server

1. If you changed the AlternateImageServerURL in the content.properties file, perform the following steps so that Content Server can communicate the alternate Image Server URL to the Workflow Server:
  - a. Restart Content Server. This writes the URL to the Workflow Server.
  - b. After Content Server has restarted, restart the Workflow Server. This forces the Workflow Web application to re-query Content Server for the alternate Image Server url.
2. Import the certificate of the CA that signed the Image Server and/or Portal Server certificate into Workflow Server:
  - For Java portals, refer to [“Importing CA Certificates into the cacerts Keystore \(for Java Portals\)” on page 3-86.](#)
  - For .NET portals, refer to [“Importing CA Certificates into MMC \(for .NET Portals\)” on page 3-87.](#)

## Setting Up Collaboration Server to Use a Secure Image Server

Collaboration Server does not require any changes to function in security modes 1 or 2, as it uses the portal's Image Server settings. A certificate is not required.

If you are using Security Mode 3, import the certificate of the CA that signed the Image Server and/or Portal Server certificate into Collaboration Server:

- For Java portals, refer to [“Importing CA Certificates into the cacerts Keystore \(for Java Portals\)” on page 3-86.](#)
- For .NET portals, refer to [“Importing CA Certificates into MMC \(for .NET Portals\)” on page 3-87.](#)

However, if the host/port of the normal Image Server URL used by browsing users is not accessible from Collaboration Server (for example, the Image Server is on a different machine than Collaboration Server), you must change the jscontrols component that Collaboration Server uses. The symptom of this problem is error messages displayed in the Calendar portlets. To avoid the errors:

1. In a text editor, open **config.xml** (located in your Collaboration Server installation directory, for example, C:\Program Files\plumtree\ptcollab\5.0\settings\config\config.xml).
2. In the following line set the URL to the value in the portal config file (j\_config.xml or n\_config.xml).

```
<jscontrols>  
  <imageServerConnectionURL>[URL]</imageServerConnectionURL>
```

## Setting Up Studio Server to Use a Secure Image Server or Portal Server

The only thing you need to do is import the certificate of the CA that signed the Image Server and/or Portal Server certificate into Studio Server:

- For Java portals, refer to [“Importing CA Certificates into the cacerts Keystore \(for Java Portals\)” on page 3-86.](#)
- For .NET portals, refer to [“Importing CA Certificates into MMC \(for .NET Portals\)” on page 3-87.](#)

## Troubleshooting

### *Running KeyTool*

- If the keytool command is not recognized, it might be because Java is not in your path. Change your directory to the Tomcat installation directory, for example, C:\Program Files\jakarta-tomcat-4.1.18-LE-j2sdk1.4.1\_02\j2sdk1.4.1\_02\jre\bin.
- If, when running the keytool command, you get an “alias already exists” error, change your command's “-alias” argument to use a different alias.

### *Content Server Errors*

The following errors indicate that the Content Server has not been set up with SSL:

- Workflow Tracker portlet displays “Unable to display this page”
- The Community Templating Style Sheets, Community Branding Image Publishing Target, and Community Branding Image Preview Target display “Write Channel Closed, possible SSL handshaking or trust failure.”
- Attempts to create a content item or branding portlet display an http status 500 error:  

```
org.apache.jasper.JasperException: jscomponent file for jscontrols
not found or failed to load. Exceptions encountered:
com.plumtree.openfoundation.io.XPIOException: java.secu-
rity.cert.CertificateException: Couldn't find trusted certificate -
com.plumtree.openfoundation.io.XPIOException: Unexpected end of file
from server
```
- Navigating to the image directory of the Community Directory Section in Content Server Explorer displays a blank page when trying to view the images.

You should be able to correct these problems by performing the steps described in [“Setting Up Content Server to Use a Secure Image Server or Portal Server” on page 3-88](#).

If Workflow portlets are not displayed when using an alternate Image Server URL, refer to Plumtree Knowledge Base article 230242.

### *Workflow Server Errors*

If the My Activities portlet displays “An unknown error has occurred”, the Workflow Server is not working. Import the certificate of the CA that signed the Image Server and/or Portal Server certificate into the Workflow Server JRE:

- For Java portals, refer to [“Importing CA Certificates into the cacerts Keystore \(for Java Portals\)” on page 3-86.](#)
- For .NET portals, refer to [“Importing CA Certificates into MMC \(for .NET Portals\)” on page 3-87.](#)

## Search Server Security

The Plumtree Search Server should always be deployed in a secure network, behind a fire-wall. The Search Server authenticates connections from the portal and servers using a fixed, private key. There are no configurable security options.

## Collaboration Server Security

There are no special security considerations with regards to Collaboration Server. The exception being if you deploy your Image Server over SSL but have a topology that does not allow the Collaboration Server machine access to the Image Server over a standard http connection. For those configurations, there is a fix in Plumtree Corporate Portal, version 5.0.2 and Collaboration Server, version 3.0.2 that allows you to configure an alternative http method for accessing the Image Server.

The problem is that the jscontrols components (used for the calendar, for example) require that the portal and server machines have access to the Image Server. In certain configurations, portal *users* have access to the Image Server over SSL but the portal and server machines do not. In order to provide a workaround for this configuration, there is a new configuration setting in the portal and Collaboration Server called the “image server connection URL,” which allows an alternate access URL for the portal and server machines that do not have access to the Image Server over SSL.

In the case of Collaboration Server, there is a new `<jscontrols>` section in **config.xml**. The `<imageServerConnectionURL>` section is blank by default, but this is where you can configure an alternate path to the Image Server for the jscontrols components.

## Content Server Security

Content Server should be deployed in a secure environment, behind a firewall. When Portal Servers and Image Servers are deployed using SSL (with security modes greater than 0), you must import certificates from the Portal Server and Image Server into the Java application servers running Content Server and the embedded workflow engine. For detailed instructions, refer to [“Configuring SSL for Your Enterprise Web Deployment” on page 3-82](#).

## Studio Server Security

There are no special security considerations with regards to Studio Server. The exception is when the Image Server is configured with a network topology that does not allow the Studio Server machine access to the Image Server over a the same http connection URL as is used by browsing users. For example, the Image Server might be located in a DMZ and not be visible over the same port as it is to users outside the DMZ. This can cause certain Studio Server components to fail. For those configurations, there is a fix in Studio Server, version 3.0.2 that allows you to configure an alternative http method for accessing the Image Server.

To configure an alternate image server URL, edit **PTStudioConfig.xml** as follows:

```
<portal>
  <studio-registrar/>
  <studio-registrar-pwd/>
  <image-server-connection-url>http://[imageserver]:[port]/image-server</image-server-connection-url>
</portal>
```

## Analytics Server Security

There are no special network security considerations with regards to Analytics Server. Refer to the *Installation Guide for Plumtree Analytics Server* for information on ensuring secure user access.



## The DMZ (Demilitarized Zone)

A DMZ (sometimes referred to as a demilitarized zone or perimeter network) is a computer or small network inserted as a neutral zone between a company's private network (intranet) and the outside public network (Internet or extranet). A DMZ uses some combination of firewalls and gateways to prevent outside users from having direct access to a server that holds company data.

## Firewalls and Security

A firewall can be a valuable component in an overall security strategy. However, firewalls alone do not create security. In fact, it could be argued that, with conscientious Web server and operating system security policies, firewalls could be dispensed with entirely. Firewalls typically provide the first line of defense, intelligently routing requests and filtering out those that do not meet requirements configured into the device (or software). Depending on the sophistication of the firewall product, more or less intelligence can be built into the decision tree affecting whether a packet should pass through the firewall. Having a firewall in place can provide a false sense of security, however. Consider the following real world scenario:

Suppose a Web server operating in a DMZ behind a firewall restricts traffic to port 443 (HTTPS) requests. A second firewall insulates the internal network from the computers in the DMZ. A hacker from the Internet sends a buffer overrun attack to the IP address of the Web server. The data looks like a regular HTTPS request, it goes to port 443, and is passed through to the Web server as a TCP stream. The Web server tries to decrypt the stream in the normal fashion. The data inside the request exploits a weakness in the Web server that allows it to overrun the memory stack of one of its threads. The thread executes some code sent by the hacker. The code gains control of the Web server, opens a new socket and sends a similar malicious request to the next server in the HTTP(S) chain. The request goes unnoticed by the second firewall (it still uses HTTPS TCP port 443). The target Web server is controlled in the same fashion. If the second server is a member of your primary domain, the hacker has a good access point to your network. If the buffer overruns were done carefully and security audits for successes (versus failures) are not implemented on the Web servers in the chain, it is unlikely the attack would even be noticed.

This is not to imply that firewalls are useless. On the contrary, firewalls can dramatically limit the nature and even the source of potential attacks. Firewalls must be supplemented by good internal security policies and measures. For example, by restricting the rights and privileges of the user in whose process space the Web server runs, risks can be minimized or eliminated. Furthermore, with careful configuration of network trusts and operating system security audits and alerts, an attack such as that described above could be very difficult to implement.

## Implementing the Plumtree Corporate Portal in a DMZ

The remainder of this section focuses on the positioning of Plumtree servers with respect to firewalls and perimeter networks (DMZs). Plumtree does not advocate the use of any specific configuration. This section presents several possible topologies that incorporate firewalls, since they are a common element of many company infrastructures.

The most important security measure is the “hardening” (establishing maximum possible security) of the computers involved in the portal configuration, especially those that receive direct user requests. These Web server computers are sometimes referred to as bastion hosts, since they are typically the most vulnerable to attack. Establishing the appropriate privileges for the plumtree user is a critical component of this activity (as is installing all of the appropriate security patches for the operating system and application server). However, once steps are taken to secure bastion hosts and other computers, you can provide extra layers of security to protect against unknown vulnerabilities in the operating system or Web server software.

### Perimeter Networks (DMZ)

A perimeter network is an untrusted part of an enterprise network that sits between internal resources and the outside world. It can consist of any number of devices and servers designed to filter, inspect, or route requests coming into the perimeter, and typically governs all traffic allowed to pass through to the internal network. In this way, all perimeter networks are alike. There are many different products, both hardware and software, that provide the filtering, inspection, and routing functionality. A company must determine the appropriate balance between performance and simplicity when implementing security. Simply put, as the number and sophistication of inspection stations increases, so does security, but, frequently, at the expense of performance and administrative simplicity.

### Enterprise Web Components and Their Communication Protocols

Understanding how the various Plumtree servers communicate with one another can help you to understand the options in implementing a secure topology. The following diagrams illustrate the protocols used between the various computers that might comprise a Plumtree deployment (.NET and Java deployments are shown).

Figure 3-1: Physical Architecture for a .NET Deployment

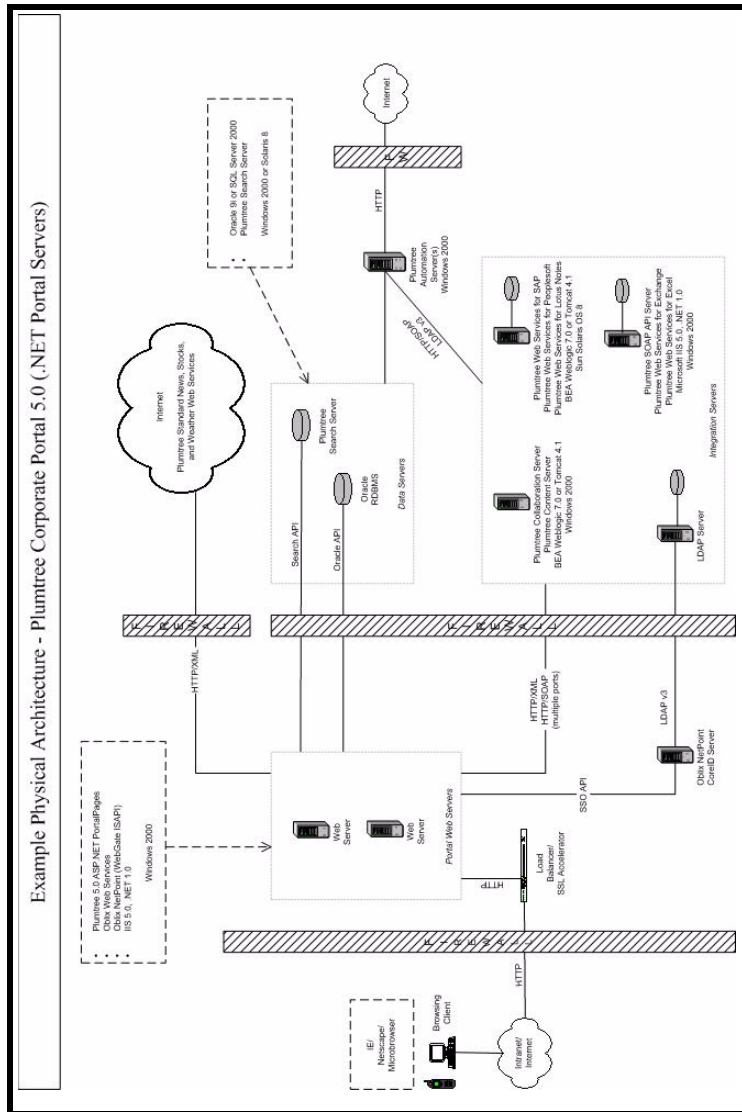
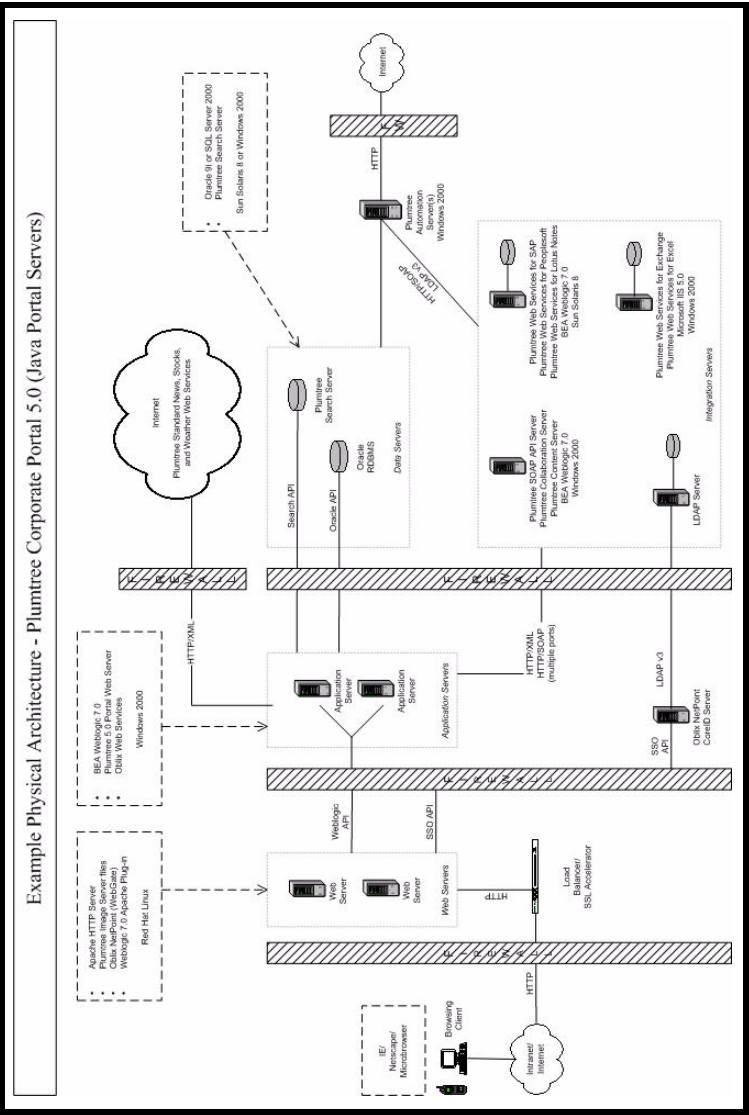


Figure 3-2: Physical Architecture for a Java Deployment



To better understand the connections in these diagrams, we will go through each server and describe the network connections, protocols, and DMZ impact of each:

**Portal Server** - The Portal Server is the single point of access for end-users using TCP/IP + HTTP-based Web browsers. A Portal Server should be placed in the DMZ, protected by a firewall and possibly a proxy server. For the Portal Server to run properly, constant connections to the Search Server and Database Server are required. The Portal Server is also configured to connect to many web services, including Enterprise Web servers such as Content Server, Collaboration Server, and Studio Server.

**Image Server** - The Image Server is not a Web application. It is simply a Web server that can serve non-secured, static .gif, .jpg, html, and JavaScript files. HTML returned from a Portal Server will reference these static files, and browsing end-users will connect to an Image Server through a standard TCP/IP + HTTP/S connection. The Image Server is commonly placed in the DMZ and will often allow FTP transfers from the internal network to update its files.

**Search Server** - The Search Server should reside in the internal network and allow connections from all the Enterprise Web servers. The port numbers are configurable in `ptsearchserver\5.0\settings\ignite.ini`.

**Database Server** - The Database Server stores all the system information of the Enterprise Web and connections are required from Portal Server, Content Server, Collaboration Server, and Studio Server. SQL Server and Oracle use different default port numbers and these can be configured.

**Automation Server** - Automation Servers should be deployed on the internal network. They require connections to the Database Server, Search Server, crawler web services, and authentication web services. Unlike previous versions, the 5.0 Automation Server stores job logs in the database, so there is no need to have a shared files server.

**WS Server** - The WS Server is a remote portal API that is called by integration web services using SOAP protocols. The WS Server should be deployed in the internal network and requires constant connections to the Database Server and Search Server. The WS Server and the Portal Server do not interconnect.

**Document Repository** - The Document Repository is a SOAP-based remote API that is called by Enterprise Web servers to centrally manage document storage. The Document Repository should be deployed on the internal network.

**Workflow Server** - The Workflow Server is deployed on the internal network as a J2EE application. Clients use a Workflow Client API to communicate with the Workflow Server via HTTP/XML protocols.

**Collaboration Server** - Collaboration Server is deployed as a Java Web application on the internal network. Collaboration Server provides HTTP-based portlet web services that are called by the Portal Servers. End-users do not connect to Collaboration Server directly. For Collaboration Server to run properly, constant connections to the WS Server, Image Server, Document Repository, Search Server, and Database Server are required. A connection to Content Server is required for the optional “submit to workflow” feature.

**Content Server** - Content Server is deployed as a Java Web application on the internal network. Content Server provides HTTP-based portlet web services that are called by the Portal Servers. End-users do not connect to Content Server directly. For Content Server to run properly, constant connections to the WS Server, Image Server, Document Repository, Workflow Server, Search Server, and Database Server are required. Content Server can be configured to deploy static content to the Image Server through an FTP connection.

**Studio Server** - Studio Server is deployed as a Java Web application on the internal network. Studio Server provides HTTP-based portlet web services that are called by the Portal Servers. End-users do not connect to Studio Server directly. For Studio Server to run properly, constant connections to the WS Server, Image Server, and Database Server are required.

**Integration Web Services (AWS, PWS, GWS, SWS, CWS)** - These are n-tier Web applications deployed in standard application servers that support development of web services (for example, Java, ASP.NET, ASP, Perl, ColdFusion). These web services are called by other Enterprise Web servers using HTTP/SOAP and HTTP/XML protocols. A typical Enterprise Web deployment may have tens or hundreds of integration web services deployed on the internal network over port numbers suitable for HTTP traffic and firewalling. Each of the web services can, in turn, call other systems of record using various protocols (for example, LDAPv3, native database protocol, IIOP, RPC, NetBIOS, SMB/CIFS).

Traffic from integration web services to systems of record is not typically firewalled since the connections are inside the internal network.

## Web Services and Internal Network Security

Residing in the DMZ, the Portal Server requires the most scrutiny in designing for security. Except for the Database and Search Server, all requests from the Portal Server into the internal network are made through web services protocols using TCP/IP and HTTP 1.1. This web services architecture provides the following security advantages:

- HTTP 1.1 is a well-known protocol for tools to monitor and audit.
- Each web service runs over a single, configurable port number, which is easy to protect with a firewall.
- The Portal Server implements the full range of HTTP security, including SSL/TLS, certificates, and basic authentication when making requests.
- Single sign-on (SSO) products that are designed to protect HTTP traffic can be used to protect web services residing in the internal network. The portal is designed to forward SSO tokens as needed.
- Full connections to systems of record are not provided to the DMZ. Each web service is designed to provide remote calls to specific functionality. This is analogous to providing only a set of stored procedures instead of full DBA-level access database to a client-server application.
- On NT networks, the portal can authenticate against multiple NT domains with no trust relationship by using multiple NT authentication web services.
- Administrators can ban all traffic except known HTTP connections from the DMZ into the internal network. Protocols considered “unsafe” for use in a DMZ are limited to use on the internal network.

Plumtree provides many out-of-the-box integration web services to connection to Windows Active Directory, LDAP Servers, Documentum, Microsoft Exchange, Lotus Notes, SAP, Peoplesoft, and Siebel, just to name a few. Companies are encouraged to develop custom web services to integrate with internal systems. From a security standpoint, these are all the same and all would be deployed on the internal network. Companies can use firewalls to further compartmentalize internal networks as needed.



## Extranet Strategies

### Authentication Source Provider Configuration

#### *NTLM (NT Challenge/Response)*

NTLM is an authentication protocol used in various Microsoft network protocol implementations. Originally used for authentication and negotiation of secure DCE/RPC, NTLM is also used throughout Microsoft's systems as an integrated single sign-on mechanism. Plumtree allows NTLM authentication, which means that a user already logged in to a Windows workstation can open a browser and be authenticated with the portal without logging in.

NTLM has advantages and disadvantages. The user enters the portal directly, but the portal is unaware of the username and password of the user. So, for example, the portal cannot forward this authentication information to remote servers for use by your custom applications. Keep in mind that you can store and manage your users in an NT or Active Directory domain without using NTLM. Users can enter their names and passwords manually, and if they enable the "Remember My Password" feature, they will only have to log in infrequently.

If you do not want to open these ports in your firewall, then you might want to consider routing your SMB traffic in another way. Direct Host, IPSec, PPTP, IPX/SPX, and other VPN tunneling protocols are all possible alternatives for routing SMB traffic through fewer or alternate ports and providing various additional security and encryption safeguards.

Alternately, you might want to bypass the issue altogether by avoiding the use of NTLM authentication.

### *LDAP Authentication Sources*

Your machine hosting your authentication web service needs to be able to connect to the LDAP machine; it does not matter if your Portal Server is in the DMZ and your LDAP directory is inside the firewall. If you are not using LDAP over SSL, then, by default, you will need to open port 389. If you are using SSL, then, by default, you will need to open port 636. Most LDAP directories allow you to change the port for your LDAP directory. If you have changed the port, then you will need to open that port in your firewall. By default, a Plumtree LDAP authentication source will try to communicate with the LDAP directory over these default ports. However, if you have changed the port, then you will need to enter the correct port in the LDAP Authentication Source Editor, on the LDAP Settings page.

### **Risk Mitigation Scenarios**

Following are some simplified examples of perimeter network topologies. In all cases, the target audience for the portal is both internal and external, thus some form of perimeter network is implemented. VPN topology is deliberately omitted, although it is a very common means of accessing internal portal content from outside the firewall. For the purposes of this discussion, VPN is considered equivalent to internal network access.

#### *Scenario 1: The Reverse Proxy + IIS*

One of the simplest implementations of a perimeter network involves a reverse proxy. Much as proxy servers route traffic from the internal network to the Internet, reverse proxies route traffic in the opposite direction. A reverse proxy can be hardware (for example, F5 Big IP Application Switch) or a software component (for example, MS Proxy Server), and can incorporate other functionality, such as packet inspection and intelligent routing. The reverse proxy generally acts as the firewall between the outside world and the internal network, but can also be used in concert with multiple firewalls. Reverse proxies are somewhat controversial among network administrators, but are commonly used by companies who view reverse proxy servers as one component in an overall solution.

### *Scenario 2: Multiple Network Cards*

A second very common practice is to create multiple networks in a single computer through the use of two or more Network Interface Cards (NICs). With this scenario, you might also incorporate a firewall in addition to multiple NICs, with the firewall and one NIC serving as the perimeter network. The firewall blocks all traffic except HTTP requests, which are received by the Portal Server on the first NIC. The Portal Server uses the second NIC to communicate with other Enterprise Web computers residing on the internal network. Multiple NICs on the other Enterprise Web computers with yet another firewall can serve to separate them from the true internal network computers. In this case, adding another layer of indirection offers considerable benefit, without any associated performance degradation. The extra network card doubles the network bandwidth of the Portal Server, improving scalability, increasing network security, and eliminating the need for server-to-server encryption, since there is no access to the dedicated subnet.

### *Scenario 3: Limited Functionality for External Users*

Given Plumtree's architecture, it is possible to deploy multiple Plumtree Web servers that implement different functionality. For example, one Web server might implement administrative functionality (for example, crawler creation), while another might not. Administrative users are redirected to a separate URL to take advantage of the administrative portal pages. Similarly, external users can be granted more limited use of the portal than internal users. (Users that access the portal through a VPN are considered internal users.) Relegating all externally accessible resources to the perimeter network can eliminate virtually all traffic through the firewall for external users. At some sites, the only resource accessible through the firewall is the database server using vendor-specific database proxy software. For example, you might not want to allow internal documents to be searched or viewed by external users. Likewise, all external users are Plumtree users, not LDAP users, since no HTTP or LDAP connection is enabled.

## Encryption

Generally speaking, encryption is the process of converting plain text into code in order to prevent any but the intended recipient from seeing that data. There are many types of data encryption, and they are a key element of network security. For portals, encryption is important in three primary contexts. These are:

- data transmitted between a user's Web browser and Portal Servers; this is the most important context, since data might pass over the Internet.
- data transmitted between one Portal Server and another. In this case, a physically secure network is the ideal solution. Secure tunnels can also be used. Encryption should only be used if physical security cannot be implemented; that is, different organizations control the servers, or they are physically separated.
- data persisted by the portal

For the first two cases, Secure Sockets Layer (SSL) provides a convenient, standard means of encrypting any data transmitted over HTTP. Plumtree uses an alternate mechanism for encrypting persistent sensitive data.

### Secure Sockets Layer (SSL)

SSL prevents potential eavesdroppers from intercepting information (such as passwords) sent to Web sites and information (such as secure documents) the Web site sends back. Encrypting and decrypting communications requires processing and increases the time required for pages to load and display. A Web site can use SSL on some pages and not others, so most sites find a compromise between performance and security by using SSL only in strategic locations. For example, most e-commerce sites allow you to browse selections and add items to your shopping cart over HTTP, then they switch to HTTPS when you check out to protect personal information, such as your name and credit card number. Although there is some overhead establishing an SSL session, the overhead for encrypting and decrypting during an established session is usually insignificant.

Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents on the Internet. SSL works by using a public key to encrypt data that is transferred over the SSL connection. All the major browsers support SSL, and many Web sites

use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with https rather than http.

In the Plumtree portal, SSL can be used to encrypt some or all of the traffic between the user's Web browser and the portal. Plumtree supports three security levels, which are implemented by setting the SECURITYMODE parameter in the configuration file (n\_config.xml for .NET deployments and j\_config.xml for Java deployments) to one of the following values:

- 0—The portal runs in HTTP (though HTTPS requests are still allowed).
- 1—The Portal runs in both HTTP and HTTPS. Pages of Activity Spaces listed in Secure-ActivitySpaces.xml (Login Page, Create an account, Change Password, and such) are HTTPS and all the other pages are HTTP for better performance.
- 2—The Portal runs in HTTPS. (The Portal redirects any HTTP requests to HTTPS). This mode is best for very secure applications for which performance is not a major concern.
- 3—This mode is the same as mode 2 with an SSL Accelerator.

SSL can also be used to encrypt traffic between Web Servers and Automation Servers and portlet, crawler, Search, or Authentication web service servers. The Parallel Portal Engine supports SSL encryption of the parallel requests made to these remote servers, allowing safe transmission of portlet preferences and other data delivered from Portal Servers to Plumtree web services servers. To implement SSL between Portal and web services servers, register the remote server (the remote server object is used for all four types of web services) with an HTTPS URL. On Unix and Linux, the PPE implements SSL using OpenSSL libraries. The SSL/TLS strength and algorithm used is determined by the negotiation between the connecting parties. On Windows, the portal engine uses Microsoft SSL libraries.

## Encryption of Persistent Data

Plumtree provides a means of encrypting sensitive information that is persisted to any of a variety of repositories (for example, the Plumtree database, the Windows registry, configuration files). The following table shows the various places encryption is used for persisting information.

Data stored by Plumtree are most commonly encrypted using a 40-Bit RC2 algorithm. The algorithm is fixed in the software and cannot be altered. 40-Bit was chosen to comply with export regulations and to provide reasonable obfuscation of data in the Plumtree database, independent of other means of encryption. It is deemed sufficient because:

- both the repository and the communication channel can be secured independently using vendor-specific techniques (for example, Oracle Advanced Networking Option and data encryption)
- the database server should live in a cold room, physically isolated from other computers, on an isolated subnet where it can only talk to other Plumtree servers.

## PKI and Digital Certificates

This section describes Public Key Infrastructure (PKI) and digital certificates as they relate to the portal. Scenarios in this section present practical examples of potential problems and their solutions.

### About Public Key Infrastructure (PKI)

#### *Public Key Cryptography*

PKI systems use two mathematically related keys known as the public key and the private key. The public key can be distributed publicly without compromising the security of a system, as long as the private key remains secret. Anyone can use the public key to encrypt a message such that only someone with access to the private key can read it. Of more importance to this discussion, someone with access to the private key can digitally sign a message. Anyone can read a digitally signed message and can use the public key to verify that it was signed with the private key, proving the identity of the author.

### *Attack 1: The Imposter and the Bank*

Suppose a bank receives an e-mail message directing the transfer of \$10,000 from a particular account to a numbered Swiss bank account. The message is, purportedly, from a bank customer and is digitally signed. The bank officers attempt to verify the digital signature, but, as they have never received e-mail from that particular customer before, they do not have the customer's public key in the system. The e-mail contains a link to a public key server, so the officials follow the link, look up the key, and use it to verify the signature. Everything checks out, so they transfer the money...to the account of the impostor who has just robbed the bank customer's account.

How? The impostor set up a key server and entered his own public key under the bank customer's name, then signed the directive using his own private key. The signature is perfectly valid since the public and private keys correspond. The bank was fooled into thinking the impostor's public key was that of the customer, making the directive seem genuine.

### *Defense: X.509 Digital Certificates*

The X.509 digital certificate standard was designed for a single purpose: to certify the owner of a public key. A certificate contains a public key and the name and contact information of its owner, all signed by a trusted certificate authority. The certificate authority has taken steps to ensure the bank customer's identity before issuing the certificate, and, verifying the authority's digital signature ensures that the certificate is not a forgery.

Returning to the example above, the impostor will be unable to obtain a certificate for his public key in the name of the bank customer from a reputable authority. If he tries to use his own certificate, the bank officials will note that the owner of the key is not the purported signer of the e-mail and block the transaction.

### *Attack 2: Using a Digital Certificate to Prove Identity*

Suppose the bank receives another e-mail directing a transfer, again purporting to come from one of its customers. This message is not digitally signed, but attached is a copy of the customer's X.509 digital certificate. The certificate contains the customer's name and

address and is properly signed by a trusted certificate authority. Should the bank officers approve the transfer?

No, not if they understand PKI. A public key is, by definition, public information. For people to verify a signature, the owner of the signature posts it to public key servers and distributes it to anyone who asks. The message could have come from anyone who has access to a computer.

### *Conclusion: Signed and Certified*

To be confident of identity in a PKI system, two pieces of information are needed:

- A valid certificate, proving ownership of the public key
- A valid signature, proving knowledge of the corresponding private key

SSL employs both of these pieces of information to provide secure client authentication as an optional part of the SSL handshake. If requested by the server, an SSL client provides a copy of its client certificate and digitally signs a digest of the handshake. An impostor can easily supply a copy of a stolen certificate, but cannot forge the signature without knowing the corresponding private key.

## Delegation and Portals

### *About Delegation*

Delegation is a technical term for a process in a computer security system that allows a system to act on behalf of a particular user, particularly when accessing other systems. As an example, consider an e-mail portlet in a portal. When the portal system attempts to access the e-mail system, the latter will respond with a challenge for credentials. At this point, the portal can prompt the user for credentials or use credentials stored earlier. What you want to be able to do, however, is delegate to the portal system the authority to access the e-mail system on your behalf. This grant of authority should last only as long as you are connected to the Portal Server and should not require storing credentials permanently. Kerberos is an excellent example of a system that permits this sort of delegation.



### *PKI Does Not Permit Delegation*

In any PKI system, the private key is jealously guarded, since access to it grants the privilege of signing messages. Since signing messages is the only way to prove identity within PKI, the only way to delegate authority is to share the private key. This is not controlled, secure delegation, but rather an unlimited, permanent grant of authority.

### *Application to Portals*

Returning to the portal example cited previously, consider the case where the portal has been configured to authenticate using the client certificate option of SSL. When a user connects a Web browser, it sends a signed message accompanied by the user's certificate, and these together prove the user's identity to the portal. Can the portal now pass the user's certificate on to the e-mail system to retrieve mail? No, for the same reason the bank officials will reject attack 2. If the e-mail system accepted the certificate as proof of identity, anyone could easily access an e-mail account using the publicly available certificate.

So, to access e-mail on a user's behalf, the portal system needs the private key, which it does not have after the SSL handshake. Browsers do not supply any automatic way to transfer this key (nor should they), so the user needs to go through some configuration process of extracting the private key from the key store on the local machine and uploading it to the portal for storage. The user would either repeat this process on every login or allow the portal to store the certificate permanently. In the latter case, the portal becomes a holding point for every certificate in the system, a sort of master key room, which can potentially grant a hacker access to every system as any user.

## Using PKI in Your Portal

### *Additional Drawbacks to Digital Certificates*

Delegation is the major problem, but digital certificates have other drawbacks as a portal single-sign-on solution:

- Digital certificates are hard for users to set up. While the Web browser might come with support for certificates, the user still needs to install a personal certificate.
- What about users accessing from multiple computers, including, for example, an airport kiosk? Either administrators must prohibit this, severely restricting functionality, or users must remember to uninstall their certificates or risk leaving their credentials installed on public machines.
- Certificates are generally valid for many days, or even years, after they are issued. This creates the problem of certificate revocation, as an organization will want to cancel a certificate before the expiration date if an employee leaves or a certificate is stolen or compromised. Doing this requires issuing and maintaining certificate revocation lists, which are not yet standardized and create administrative and performance issues.

### *Complete Solution: PKI with an SSO Server*

The Plumtree OEM version of Oblix NetPoint SSO software supports the use of digital certificates to securely and transparently authenticate against the SSO server, which then issues a delegable credential (in the form of an HTTP cookie). Using out-of-the-box configuration options, the portal can accept this credential for user authentication and forward it to selected portlet web services. Further, the Oblix NetPoint product provides tools to simplify the process of issuing and managing certificates and can be configured to accept other forms of authentication (for example, passwords) for users on the road without access to their desktop certificate. To implement this option, consult the Oblix documentation to enable and deploy PKI within that system.

Plumtree also supports integration with SSO products from other partner vendors.

### *Stand-Alone Solution without Delegation*

In the absence of a cookie-based SSO solution, there are two approaches to configuring the portal to accept client certificates for authentication. Both require using SSL on the login page (security modes 1, 2, or 3). When using certificates tied to Windows domain accounts, you can configure IIS to accept client certificates and then use the out-of-the-box NT SSO configuration on your portal. This is the easiest approach to take since it leverages the built-in features of Windows, and is the recommended approach whenever possible. To accept certificates from users unrecognized by IIS, you will need to implement a custom SSO solution, which entails writing a custom SSO vendor class in Java or C#.

### Summary

You can install server digital certificates on any Plumtree Server (Administrative Portal Server, Portal Server, Remote Servers, and so on) and enable SSL. Plumtree supports SSL between browser and server as well as between Portal Server and Remote Servers.

You can use a digital certificate and SSL to communicate with your LDAP server.

You can use client digital certificates with SSL to authenticate users to the portal. This can be done with Windows Integrated SSO, with a custom SSO solution, or in conjunction with a supported third-party SSO product (for example, Netegrity, Oblix).

The portal *cannot* “pass-through” digital certificates to Remote Servers, because this is impossible, because SSL does not permit delegation.

You can do SSO to portlet web services and use digital certificates to log in to the portal, but only if you use a third-party SSO product that supports both cookie-based SSO and digital certificates (this includes Oblix WebGate and Netegrity SiteMinder). In this case, users use the digital certificate to log in to the SSO server and obtain the SSO cookie, Plumtree accepts the SSO cookie and forwards it to portlet web services.

## Single Sign-On Options

### What SSO Means in the Enterprise Web

Single sign-on (SSO) has many different meanings in different contexts. It can mean protecting your Web server with a product from an SSO vendor, such as Oblix. It can mean preventing your users from having to enter credentials more than once, or at all. It can mean identity management or a way to store credentials for many systems to simplify user experience and administrative management. Usually it means some combination of these notions.

The key to understanding your SSO requirements is realizing that the portal is based on a loosely coupled architecture; different tiers of components communicate with each other, primarily over HTTP. The end-user connects to the portal tier over HTTP; the portal connects to numerous other systems over HTTP, including many systems that act on behalf of the end-user. The key is to make this simple for the end-user, simple enough for the administrator, and secure enough for the security team.

### Logging in to the Portal

#### *Delegating to Remote Authentication or SSO*

When users point their browsers at the portal, the default experience is for users to be presented a login screen. This login screen allows users to authenticate against any authentication source, which might be a remote system such as LDAP, the portal database, or an SSO provider.

When delegating authentication to an LDAP authentication source, the portal can be configured to keep the user credentials in a safe location within memory for later use. The sequence of events for LDAP is as follows:

1. User goes to portal HTTP address; enters credentials
2. Portal stores credentials in safe section of memory
3. Portal sends request to LDAP authentication source

4. LDAP authentication source returns OK
5. User is granted access to their profile in the portal

Delegating authentication to an SSO source can circumvent the Plumtree login screen and engage the end-user in the SSO login mechanism (could be a login screen, a key card, or some other mechanism). Common SSO sources include Oblix, Netegrity, and Windows Integrated Authentication (WIA). The sequence of events for Oblix would go something like this:

1. User goes to portal HTTP address
2. Oblix intercepts this HTTP request, realizing user is missing a cookie. If user already has cookie, skip to Step 6.
3. Oblix redirects to Oblix server
4. Oblix server authenticates; sets browser cookie
5. Oblix redirects to original HTTP address
6. Oblix intercepts this HTTP request, recognizes valid cookie and instructs Plumtree to grant user access to their profile

In both cases the authentication was delegated to an external source. In both cases this was likely ultimately delegated to LDAP. In the first case, however, the portal has the user's password for later use (if configured to do so). In the second case, the SSO vendor might have employed any of several authentication mechanisms that it supports, whether login screen or keycard.

## Logging In to the Portal with Auto-Authentication

### *WIA and Advanced Mechanisms*

A user might log in to the portal without ever entering login information. On Plumtree's Windows product, for example, Windows Integrated Authentication (WIA) SSO is available (including both NTLM and Kerberos5), while different SSO vendors support different zero-login features.

## WIA

When using Windows Integrated Authentication, the user must be logged in to a machine on a Windows network. The browser on this machine is smart enough to pick up the user's identity, so the browser can negotiate with the portal to establish the users credentials. The sequence of events for WIA is as follows:

1. User logs onto a Windows network; opens a browser
2. User goes to portal HTTP address
3. Portal challenges the browser with an WIA challenge (401 Unauthorized)
4. The browser asserts an encoded piece of information (Negotiate)
5. The portal challenges the browser with a piece of information (Challenge)
6. The browser asserts another encoded piece of information (Response)
7. WIA accepts the user's HTTP request; if the credential were incorrect, the user would be challenged with a login screen in Step 6
8. The portal accepts the user's identity brokered by WIA and grants user access to their profile

Notice that the portal was never able to capture the user's password, and the user needed to be logged in to a Windows network for the authentication to succeed. Additionally, a multi-pass handshake occurred between the browser and the portal. Companies often request that the portal be able to repeat this WIA authentication between the portal and the remote server. The portal cannot do this, because there is no forwardable token; Although WIA supports not only NTLM but also Kerberos5, which theoretically supports delegation, no supported browsers implement delegation. So, not only is the portal unable to broker this multi-pass handshake, WIA will fail across any HTTP proxy.

Other important considerations when using WIA are that the user must be an Active Directory or NT user, Internet Explorer or Netscape 7.1+ must be used, and proxies between the browser and portal are not allowed.

## *Other Advanced Mechanisms*

Different SSO vendors enable different authentication mechanisms; discuss this with your SSO vendor for full details. One such option is a key card; with the card inserted into their machine, a user can authenticate without being shown a login screen.

## Brokering Credentials to the Remote Tier

After the user has logged in to the portal, the portal wants to serve the user exciting Enterprise Web applications. These applications might be pulling from custom systems as well as enterprise systems such as SAP. Let us discuss how the portal can connect to SAP.

Plumtree provides an SAP Portlet Framework that allows business users to provide SAP functionality in their applications. This SAP framework is a remote component that calls into various SAP APIs. These APIs require a username and password. By default, the framework will give user credentials to the SAP system.

Plumtree can pass user credentials in any of several ways:

- **Preferences:** Users can set credentials as preferences, and these preferences can be sent in the HTTP request. This is useful if SAP runs off of different credentials than the main network and if no mapping exists between SAP credentials and the main user credential (such as LDAP). Preferences are stored in the portal database (encrypted), and controlled by the end-user.
- **UserInfo:** The administrator can map properties from LDAP or another system into user profiles, and segments of the user profile can be sent as UserInfo. If LDAP contains entries for SAP credentials for each user, then the user will not need to ever enter their SAP credentials, but the LDAP system will need to be the system of record.
- **PassThrough:** The login credentials the user provided at login can be sent to the remote tier as a Basic Authentication header. This is very useful if both the portal and SAP are based on a user's LDAP credentials. In this case, all communication between the portal and the remote tier should be over a secure channel (such as SSL) to protect the user's password. For this method to work, the password had to be captured on login, which was not the case with the NTLM or Oblix examples described previously.

- **SSO:** An SSO token can be forwarded to the remote tier. This will not, however, let the SAP framework log in to SAP unless the SAP API accepts the SSO token as a valid user login. The SAP API that Plumtree uses (JCO) is not enabled to accept tokens of third-party SSO systems, such as Oblix. Systems such as Oblix can be used in conjunction with the SAP framework, but Preferences, UserInfo or PassThrough will need to be used to log in to SAP. It may be necessary, however to forward the SSO token if the remote server has been protected by Oblix; in this case, without the SSO token, the portal's request will be rejected. Also, the SSO token can be used with an SSO vendor's API to reconvert to name and password, but this is highly dependent on the SSO vendor and the particular company configuration of the SSO software.

## Summary

When you are evaluating SSO options for your Enterprise Web, you need to ask yourself several questions: Are you looking for single sign-on, Web property protection, or both? What Web properties do you need to protect, and to what degree? What are all of the secured systems that you need access to? What will your network topology look like (locations of browsers, Portal Servers, remote servers, as well as firewalls and proxy servers)? What categories of users are you serving (LDAP, NT, Active Directory)? Is there an SSO solution that will handle all of these systems out of the box? If not, what combination of the mechanisms described here will you use to achieve the desired effect?

The following table summarizes some features of different SSO options. This is meant as an overview; remember that many of these options can be used in combination. When determining your SSO strategy, it is wise to speak to Plumtree Professional Services or a certified Plumtree partner.



Key to Table 3-13:

✓ = out of box

C = custom work required

X = not supported

*Table 3-13: Features of Different SSO Strategies*

Feature	No SSO	Oblix	Netegrity	WIA	Custom
Can use windows identity	X	✓	✓	✓	C
Auto-login to all systems	C	C	C	C	C
Remember password	✓	✓	✓	✓	C
Forward password to remote tier	✓	C	C	X	C
Forward SSO token to remote tier	X	✓	✓	X	C
Supports X.509 client certificates	X	✓	✓	✓	C
Supports two-factor authentication	X	✓	✓	X	C
Supports non-windows applications	✓	✓	,	X	C

## Architecture for Development, Staging, and Production Environments

### Portal Server Migration

For information on migrating portal objects, refer to the *Administrator Guide for the Plumtree Corporate Portal*.

### Search Server Migration

There are two staging scenarios to be addressed:

- In scenario one, the portal is being deployed live for the very first time. In this case, Plumtree recommends that only the portal database, Web server, and so on, *not* the Search Server be ported directly from the staging environment to production. The production Search Server should start out as a freshly installed Search Server with an empty index. A Search Repair should then be scheduled, using the Search Server Manager utility in the portal. The next run of the Search Update Agent will ensure that all portal objects and documents get indexed by the Search Server. Search Repair indexes documents at roughly 2.8 documents per second or 10,000 documents per hour. This rate can vary depending on the source of the documents, the size of the documents, the format of the documents, the speed of the network, and other factors.
- In scenario two, the portal is already live, and some new portal objects are tested out in a staging environment and then pushed out onto production. No extra action needs to be taken in order for the new objects to be indexed. The Search Update Agent will soon run (it is typically scheduled to run every 10 minutes) and index the new objects for searching.

## Remote Server Migration

When migrating objects from a development (source) to production (target) portal, Plumtree recommends that existing remote servers not be migrated. The reason is that the remote server contains the base URL of the objects that depend on the remote server. In most companies, the base URL of a remote server in a development system is different than the base URL of a remote server in a production system. Therefore, if the remote server is migrated from a development to production system, the base URL of the remote server on the production system will be overwritten by that of the development system. This is the default behavior of the migration utility.



**Note:** If you select the “Export all dependent objects” option during the creation of the migration package (export), any object requiring a remote server will have its remote server included in the package. For example, if a portlet is migrated and that option is selected, the portlet's web service and remote server will be included the package. Since its source and target systems generally have separate URLs for remote servers, the remote server URL should not be migrated and therefore the default behavior is to not migrate existing remote servers.

However, if the development and production system share the same remote server and the base URL of that remote server changes, or if you are receiving an update of a component from an external source such as Plumtree Software, the remote server *should* be migrated. To allow this, the Plumtree Migration Utility allows you to specify whether you would like the remote server to be migrated as well.

By selecting “Overwrite existing Remote Server objects” in the migration utility, any existing remote servers in the target system will be overwritten by the settings of the corresponding remote server in the migration package.

If a remote server in the migration package does not exist in the target system, that remote server will always be created in the target system regardless of whether the “Overwrite existing Remote Server objects” is selected.

## Collaboration Server Migration

Collaboration Server migration is handled by the Plumtree Migration Utility. Refer to the *Administrator Guide for the Plumtree Corporate Portal* for more information.

## Content Server Migration

You can migrate portal objects as well as Content Server sections from one fully operational Portal/Content Server installation to another. This is commonly used to move a Content Server Published Content Portlet, such as a news article, from a development environment to a production environment, which requires a coordinated migration of both portal and Content Server objects.



**Important:** Do not export Content Server portal objects other than the Content-created portlets described above. Specifically, do not export objects contained in the Content Server administrative folder such as the Content Server remote server, Content Server web services, or administrative portlets like the Content Administration portlet or the My Activities portlet. Those objects are created during Content Server installation and contain settings specific to each installation. Importing these objects into another Content Server installation will replace the existing ones and cause the target environment to fail to function properly.

**Important:** Do not choose the Export All Dependent Objects option. This will cause each selected portlet's web service and remote servers to be exported as well. For Content Server created portlets, these objects contain installation-specific settings and already exist on the target portal. Replacing them will cause the target environment to fail to function properly.

**Important:** If you have installed the full Content Server (not the Branding Engine) version 5.0 or 5.0.1, you might see two remote servers on this screen (Branding remote server and Content Server remote server) with the same URL. This is because in versions 5.0 and 5.0.1 Branding portlets used a separate remote server object from the one used for the rest of Content Server objects, though in the end both used the same actual server. But for the purposes of migration, they are treated as though they were separate servers. Branding portlets will be exported

using the Branding remote server and Published Content Portlets will be exported from the Content Server remote server. Any other sections you want to export should also be exported using the Content Server remote server.

**Important:** You can migrate content only for Branding portlets.

Collaboration Server migration is handled by the Plumtree Migration Utility. Refer to the *Administrator Guide for the Plumtree Corporate Portal* for more information.

## Logs

For more information about an export or import, consult the logs of the Migration Wizard and Content Server.

The Migration Wizard log is located in the portal installation directory, in the same directory as the wizard, for example, C:\Program Files\plumtree\ptportal\5.0\bin\native. It contains information about the portal objects that were migrated, as well as a record of the Content Server migration.

The Content Server log file, **pcs.log**, is located in the Content Server's settings\logs folder. It contains more specific information regarding the Content Server objects that were migrated.

## Studio Server Migration

Studio Server supports migration of user created Studio Server portlets. This is especially useful in moving Studio Server portlets created in a development or staging environment over to a production environment.

To export a Studio Server portlet, an administrator simply needs to use the Plumtree Migration Utility (described in the *Administrator Guide to the Plumtree Corporate Portal*) to select the desired portlet, and export it to a .pte file. This file can then be copied from the source portal to the destination portal. There, the Plumtree Migration Utility can be used to import the .pte, which creates the portlet on the portal, as well as creates the underlying Studio Server application and database.

Several caveats are worth noting with respect to migrating Studio Server portlets:

- It is assumed that the source portal uses its own Studio Server and that the destination portal has its own (separate) Studio Server.
- The portal and Studio Server versions should be the same at the source and destination.
- It is not necessary to include dependencies when migrating Studio Server portlets.
- Any data in the source Studio Server portlet's database will not be migrated. It is possible, however, to use Studio Server's data export/import functionality to move data from one Studio Server portlet to another.
- If multiple Studio Server portlets share the same underlying database, Plumtree recommends that all of these portlets be migrated in a batch, rather than individually, so as to maintain the relationship between the portlets and their shared database.

## Analytics Server Migration

Plumtree recommends that the Analytics Server remote server not be migrated but freshly installed in a production environment.

# Internationalization

## Multilingual User Interface

The Plumtree Corporate Portal user interface is localized into **9** languages:

- English
- French
- German
- Italian
- Portuguese
- Spanish
- Japanese
- Korean
- Simplified Chinese

Each portal user can choose their preferred language by changing their locale under **My Account | Edit Locale Settings**. For example, if a portal user changes their locale setting to any of the German locales (Austria, Germany, Luxembourg, or Switzerland), the user interface language will change to German.

## Unicode Support

The Plumtree Corporate Portal is fully Unicode enabled in the Plumtree server components, the Plumtree user interface components, the Plumtree search components, and the portal database. The Plumtree user interface uses the UTF-8 encoding of Unicode when delivering HTTP content to the browser. Plumtree has standardized on UTF-8 so that the portal can display all the world's languages at the same time on any page.

## Localizing Names and Descriptions of Objects Stored in the Database

You can localize names and descriptions of objects stored in the portal database. For example, if you have created a portlet with the name “Travel Portlet”, you can give that portlet an associated German name of “Dienstreise Portlet”. The German name will display for users who have chosen German as their user interface language.

Names and descriptions can be added or modified using the administrative user interface for each object. To add a localized name or description, open the object in the administrative editor and, on the Properties and Names page, specify the localized name and description and choose the appropriate language. For more information, refer to the online help.

The portal also provides an administrative utility called the Localization Manager to export and import localized names and descriptions in bulk. The Localization Manager can be used by a system administrator who wants to translate a large number of names and descriptions at once. For example, if you wanted to translate all the portlet names and descriptions into French, German, and Italian, you could download an XML file containing the names and descriptions of all the objects in the Plumtree system, edit the list, and then upload it back into the portal, effectively replacing all localized names and descriptions for all the objects in the portal database. (If an object is not set to support localized names, it is not included in the names and descriptions that are downloaded.)

Here is a small sample of a downloaded Names and Descriptions .xml file:

```
<localizationtable>
  <languages count='9'>
    <language>de</language>
    <language>en</language>
    <language>es</language>
    <language>fr</language>
    <language>it</language>
    <language>ja</language>
    <language>ko</language>
    <language>pt</language>
    <language>zh</language>
  </languages>
```



```

<segments count='554'>
  <segment stringid='0' itemid='1' classid='2'>
    <source language='en'>Administrators Group</source>
    <target language='de'>Administratorengruppe</target>
    <target language='en'></target>
    <target language='es'>Grupo Administradores</target>
    <target language='fr'>Groupe d'administrateurs</target>
    <target language='it'>Gruppo Amministratori</target>
    <target language='ja'>管理者グループ</target>
    <target language='ko'>관리자 그룹</target>
    <target language='pt'>Grupo de administradores</target>
    <target language='zh'>管理员用户组</target>
  </segment>
  <segment stringid='1' itemid='1' classid='2'>
    <source language='en'>Plumtree Administrators Group</source>
    <target language='de'>Plumtree-Administratorengruppe</target>
    <target language='en'></target>
    <target language='es'>Grupo Administradores de Plumtree</target>
    <target language='fr'>Groupe d'administrateurs Plumtree</target>
    <target language='it'>Gruppo Amministratori Plumtree</target>
    <target language='ja'>プラムツリー管理者グループ </target>
    <target language='ko'>Plumtree 관리자 그룹</target>
    <target language='pt'>Grupo de administradores Plumtree</target>
    <target language='zh'>Plumtree 管理员用户组</target>
  </segment>
  ...
</segments>
</localizationtable>

```

The Localization Manager uses XML so that the translations for all names and descriptions in all languages can be kept in one file. The “languages” element lists all the user interface languages in the portal. The “segments” element indicates the number of names and descriptions in the portal. Finally each “segment” element contains one name or description in the portal. The source element contains the source text for translation and the translated text is stored in the “target” element for each target language. (Languages are identified using standard ISO 639-1 two letter language identifiers.)

## Adding a User Interface Language to the Portal

There are two types of portal languages: user interface languages and search languages. The portal includes 9 user interface languages and can be easily extended to support additional languages. Our supported search languages (62) are hard-coded and not extensible.

The portal user interface languages are automatically detected when the portal is started by detecting the language folders that exist under the root folder, for example, C:\Program Files\plumtree\ptportal\5.0\settings\i18n.

A user interface language can be easily added by creating a new language folder (using the ISO-639-1 language code). For example, if you wanted to add Dutch as a user interface language, you would copy the “en” language folder under the i18n root folder and rename the folder “nl”. After you do this and restart the portal, you will notice that you can select Dutch as your language in the **My Account | Edit Locale Settings** page. If you do this, however, you will notice immediately that you are missing the style sheets for Dutch. The additional elements required for a complete portal localization include:

- Style sheets
- Online help
- Javascript language files

## Adding Language Style Sheets

If you are adding a user interface language to the portal, you need to add the corresponding style sheets for that language. The CSSMILL was designed to make adding languages and generating all the language style sheets relatively easy. The folder **\ptimages\tools\cssmill\prop-text** is where all the language files are kept. Each language file in the **prop-text** folder has language specific values for font style, font size, text style, and so on. This design makes it easy to change the default font for each language. For example, if you want the default font for the Japanese user interface to be Tahoma, then you can add Tahoma to the **ja** language file in the **prop-text** folder. Besides adding a language file, you must also edit the **build.xml** file to generate the new language style sheets.

For example, suppose you wanted to add “Dutch” as a portal user interface language. Here are the precise steps to follow to add the Dutch style sheets:

1. Navigate to the `\ptimages\tools\cssmill\prop-text` folder on the Image Server. Copy one of the existing files to the same folder and rename it using the language conventions in ISO-639-1 and ISO-3166. For example for Dutch, we would rename the file to be “nl”.
2. Open the new file in a text editor and make any necessary modifications for the new language. For example, if you want to add a new default language, you could change the line

```
font.largest=20px verdana,arial,Helvetica,"sans-serif"
```

to

```
font.largest=21px Tahoma,"MS PGothic",Verdana,"sans-serif"
```

Be sure to add the new font for each font attribute in the language file.

3. Navigate to the `\ptimages\tools\cssmill\prop-color` folder on the Image Server. Edit every one of the existing color properties files and add the translation for the name of the color for the new language. For example, edit the file **color.l.properties**, copy the last `colorscheme.name` entry. Change the name according to the new language ID chosen in Step 1. In this example, we would copy and edit the line

```
colorscheme.name.zh=\\u6DE1\\u7D2B
```

to

```
colorscheme.name.nl=Lavendelblauw
```

4. Modify the Ant build script (**build.xml**) to add the new language to the style sheet collection by following the steps below. (This is the only way the script knows to create versions of the new style sheet for each of the languages supported by the portal.)
  - a. Navigate to the `\cssmill` directory and open the **build.xml** file in a text editor.
  - b. Add an entry for the new language within the **make\_main\_css** target: Copy the last `<antcall target="make_main_language_css">` entry and paste it at the end of the list. Modify the `<param name="LANGUAGE" value="pt"/>` tag by changing the value (“pt”) to the language id used in the name of the new language file created in Step 1 above. In this example, the new language ID for Dutch is “nl”.

- c. Add an entry for the new language within the **make\_508\_css** target: Copy the last `<antcall target="make_508_language_css">` entry and paste it at the end of the list. Modify the `<param name="LANGUAGE" value="pt"/>` tag by changing the value ("pt") to the language id used in the name of the new language file created in Step 1 above. In this example, the new language id for Dutch is "nl".
  - d. Add an entry for the new language within the **make\_comm\_color\_css** target: Copy the last `<antcall target="make_comm_lang_color_css">` entry and paste it at the end of the list. Modify the `<param name="LANGUAGE" value="pt"/>` tag by changing the value ("pt") to the language id used in the name of the new language file created in Step 1 above. In this example, the new language id for Dutch is "nl".
  - e. Add an entry for the new language within the **append\_index\_for\_color** target: Copy the last `<concat destfile="${INDEX}" append="true">main-style${COLOR}-pt.css=${colorscheme.name.pt}</concat>` entry and paste it at the end of the list. Change the language ID in the new line to the new language id. In this example, change the language ID "pt" to the new language id for Dutch "nl". The new line would look like this:
 

```
<concat destfile="${INDEX}" append="true">mainstyle${COLOR}-nl.css=${colorscheme.name.nl}</concat>
```
  - f. Save the file and close it.
5. After the build script modifications have been made, create the new style sheets by running the **make\_all** batch file. (See the directions in the previous section.)
  6. Verify that the new language style sheets were created based on the new language property file. Navigate to the **cssmill\css** directory and make sure that there are 20 new style files with the new language ID you chose in Step 1 (that is, `mainstyle-nl.css`). For further verification, open the **community-themes.txt** file (in the **\css** directory) and confirm that there is a new entry corresponding to the language ID used in the new language property file.
  7. After confirming that your changes are correct, move the new style sheet files from the **\cssmill\css** folder to the **\ptimages\imageserver\plumtree\common\public\css** folder.

8. Restart the Java Application Server. (This is necessary because the community-themes.txt file is loaded during Content Server startup and stored in memory.)

## Adding an Online Help Language

The portal online help is localized into the core Plumtree languages: English, French, German, Italian, Portuguese, Spanish, Japanese, Korean, and Simplified Chinese. The online help files are stored on the Image Server under the root folder, for example, C:\Program Files\Plumtree\ptimages\imageserver\plumtree\portal\private\help

The portal online help system is compiled using RoboHelp Office 2002. Each language folder contains a separate help project for that language. All the Western European language projects are compiled using the standard English version of RoboHelp. The Asian languages are compiled using the corresponding Asian language version of RoboHelp.

## Adding Javascript Language Files

The portal has several user interface components written in client-side Javascript. These components also contain user interface text messages. These string files must also be localized when adding a language to the portal.

The Javascript components are located on the Image Server. The Javascript component string files are located in the portal installation directory (for example, C:\Program Files\plumtree), in these folders:

- \ptimages\imageserver\plumtree\common\private\js\jscontrols\120449\strings
- \ptimages\imageserver\plumtree\common\private\js\jsdatepicker\118523\strings
- \ptimages\imageserver\plumtree\common\private\js\jsutil\118981\strings

The Plumtree convention for Javascript string files is somewhat different than for XML files. Instead of placing each language file in a separate folder, we have given each language file a suffix consisting of a dash and the language code. For example, to create a language file for Dutch, you would copy the English file and replace the “-en” suffix with “-nl”.

## Language Support in the Knowledge Directory

The portal search index is stored in Unicode (UTF-8) and supports 62 languages in total. The Plumtree Search engine supports advanced stemming and tokenization for 23 languages and basic tokenization for an additional 39 languages. The 23 languages supported for advanced stemming and tokenization are:

- |                        |                         |                      |
|------------------------|-------------------------|----------------------|
| • Chinese (Simplified) | • Chinese (Traditional) | • Czech              |
| • Danish               | • Dutch                 | • English            |
| • French               | • Finnish               | • German             |
| • Greek                | • Hungarian             | • Italian            |
| • Japanese             | • Korean                | • Norwegian (Bokmål) |
| • Norwegian (Nynorsk)  | • Polish                | • Portuguese         |
| • Romanian             | • Russian               | • Spanish            |
| • Swedish              | • Turkish               |                      |

# 4 Business Implementation

This chapter is written for business sponsors (or the people tasked with encouraging portal adoption) and Enterprise Web administrators. It describes best practices for putting your Enterprise Web plan into action. You might also find additional best practices in [Chapter 2 “Business Foundation”](#), in the online help, or in the administrator guide for the component.

## Crawler Best Practices

As mentioned in [Chapter 2 “Business Foundation”](#), crawlers are used to catalog and index documents and data from systems that are external to the Plumtree Enterprise Web. Plumtree provides pre-packaged crawlers for ubiquitous sources of enterprise content such as Microsoft NT file systems, Microsoft Exchange, Lotus Notes, Documentum, and Novell.

### Plumtree Crawler Web Service for Microsoft Exchange

Most enterprise e-mails tend to have limited scope in terms of the intended audience. However, a small fraction of these e-mails contain information that is suited for consumption by a wider audience. An example of an e-mail with limited scope is that of a sales manager asking a sales representative for the status of an account. There is no point in making the sales representative’s response available to anyone else. However, when a product manager, in response to a sales representative’s e-mail, provides a detailed explanation of how a critical feature in a product works and that information is not available elsewhere, the product manager might want to publish the response in the portal’s Knowledge Directory. The intent in this case is to make the information available to the entire sales force.

How does that get accomplished? One easy way to make this happen is to set up an e-mail account on Exchange called Product Information and have the Product Manager cc the e-mail to Product Information. The Plumtree Crawler Web Service for Microsoft Exchange (Exchange crawler) can be set up to crawl e-mails sent to Product Information and index them in the Knowledge Directory. If needed, access to these e-mails can be restricted to the Marketing and Sales groups. This implies that when Marketing or Sales searches by the

appropriate keyword, for example, “security,” security related e-mails sent to the Product Information account on Exchange will be available as search results.

While this example is specific, you can easily imagine this type of scenario playing out within many companies and departments.

*Best practice:* It is usually a good idea to have a content manager approve all such ad-hoc e-mails crawled in from Exchange and even edit them for grammar and content. Anything that is published in the Knowledge Directory should meet certain minimum standards of presentation.

*Best practice:* In the long run, it is important to periodically convert ad-hoc indexed content in the Knowledge Directory to regular, published content that has gone through a more formal approval cycle. Since e-mails and sources of ad-hoc content tend to accumulate over time, it is a good idea to consolidate the information and publish them via more regular channels in repositories such as NT file systems or Enterprise Content Management systems such as Documentum. Subsequently, the e-mails can be purged from Exchange, which automatically results in their being dropped from the Knowledge Directory. This type of housekeeping is important for two reasons: e-mail tends to accumulate over time, cluttering up the Knowledge Directory, and the information that they contain may be out of date.

## Plumtree Crawler Web Service for Lotus Notes

Lotus Notes e-mails can also be made available to a wider audience in a manner similar to that of Exchange (discussed in the previous section). In addition, Notes supports discussions and databases. The Plumtree Crawler Web Service for Lotus Notes can be used to catalog and index these in the Knowledge Directory.



## Plumtree Crawler Web Service for NT File Systems

This is perhaps the most used crawler. Almost every enterprise of any size tends to have networked file systems, the contents of which are shared by various groups of enterprise users. This is one of the simplest ways to share content across the enterprise.

One of the simplest use cases for crawling files into the portal is to make available certain files to extranet users, such as customers and field personnel. For example, you might want to make detailed technical documents or calibration documents or articles written by company experts available to customers as part of online customer support. In this case, just the portion of the Knowledge Directory that contains the links to the documents cataloged and indexed by the Plumtree Crawler Web Service for NT File Systems (NT file crawler) can be exposed to the customer in a content snapshot portlet. This implies that customers will only see the contents of that folder inside that portlet, eliminating all other search results that might otherwise clutter their search. Of course, these documents will also be available as part of generic searches.

The NT file crawler can be set up such that access to documents are restricted to those portal users who have access to them in the NT file system.

## Plumtree Crawler Web Service for Documentum

Documentum has traditionally been used heavily in process intensive and regulated industries such as pharmaceuticals and health care. Lately, with the birth of enterprise content management as a distinct software vertical, more and more companies are using Documentum as the system of record to store all kinds of important documents such as customer contracts and HR policies. Thus, Documentum is an important repository of finished content that needs to be accessed by stakeholders ranging from employees to customers to the company's board of directors to suppliers.

The Plumtree Crawler Web Service for Documentum (Documentum crawler) enables the cataloging and indexing of documents from specific Documentum docbases. Both mirrored and non-mirrored crawls can be set up. A mirrored crawl sets up folders in the Knowledge Directory to *mirror* the directory structure that is present in Documentum. Once again, user and group related access information from Documentum is transferred over to the representations of these documents in the Knowledge Directory, enabling secure access.

Often, the Documentum crawler is used in conjunction with Documentum portlets, which provide high-level Documentum functionality such as the ability to check out, edit, and check in documents.

## Building Crawlers to Other Enterprise Systems

Given the sheer number of packaged and homegrown applications present in the enterprise, it is not possible for Plumtree to provide packaged crawlers for every situation. Nevertheless the need for information in such systems to be cataloged and indexed is very real. This is why the Plumtree Enterprise Web Development Kit has a crawler web service, which enables the portal to systematically query any back-end repository.

How is this possible? The Plumtree Corporate Portal knows how to interact with a crawler web service and fill up the Knowledge Directory based on these standard interactions. The crawler web service in turn provides standard connection points or interfaces for any integration code that bridges to the back-end repository. This integration code uses the back-end repository's API to get data.

So, the Plumtree Corporate Portal (and Knowledge Directory) connects to the Crawler Web Service, which connects to the integration code for the back-end system, which connects to the back-end system itself.

Thus, a crawler web service along with the integration code can be thought of as a systematic way of normalizing the enterprise's repositories so that the portal can catalog and index them.

Plumtree is working with other enterprise software companies towards eliminating or minimizing the need for such integration code. If each packaged back-end repository provided web services similar to that of a crawler web service, there would be less need for custom integration code.

In the meanwhile, it is worth emphasizing that the crawler web service provided as part of Plumtree's Enterprise Web Development Kit takes care of the bulk of the connectivity related issues, complexity, and drudgery of writing custom crawler web services. Also, more and more content providers ranging from Documentum to Stellent to Microsoft Content Management System have published APIs to enable integration.

Enterprise data repositories such as Siebel, SAP, and Peoplesoft also provide APIs, some of them via web services, to query their systems. Siebel, in particular, provides very effective APIs to facilitate the building of crawler web services.

In summary, both packaged and custom crawlers enable enterprise users to access information spread throughout the enterprise's many systems. The user benefits immensely from the ability to search all systems at once and retrieve useful data, without having to worry about the originating system or the complexity of mastering their features and user interfaces. With crawler web services feeding the Knowledge Directory, it can play a central role in helping an enterprise achieve its knowledge management objectives.

## Portlet Best Practices

For detailed information on building portlets, refer to the *Enterprise Web Development Guide*.

Here are some things to keep in mind when building portlets:

- Span multiple systems, especially on search.
- Keep data fresh with date/time stamps.
- Recognize users or prompt for password in the portlet; do not save the password as a preference.
- Get user feedback and keep improving the portlet.
- Add new portlets to users' default page.
- Include failure messages, help desk, author, and version number.
- Use community, user, administrative preferences, and import user information from LDAP.
- Use the GCC to pass events between portlets and react to page-level browser events.
- React to user information like time zone, user locale, and other LDAP attributes.
- Set HTTP caching headers to public.
- Use the EDK to parse headers (gatewaying URLs on your own creates upgrade risk).
- Think about the entire process, especially where to store and how to update data.
- Use remote servers: If you move portlets or have a policy of changing URLs for security reasons, you should use remote servers to specify the base URL for portlets. This makes it much easier to move portlets around. Otherwise you would have to change the URL on each individual web services that uses the base URL. In fact, it would be very difficult to migrate portlets from development to production systems that use different portlet servers if you do not use remote servers.

## Subportal Best Practices

- Since navigation is global, be cautious since some use more screen real estate than other navigation schemes. For example, if you apply the vertical navigation and use three column navigation in a community, users see four columns, which might be too wide for many monitors.
- Make sure that the home target of subportals are accessible by users in that subportal. For example, if a home target for subportal P is community C, make sure that users in subportal P have at least Read access to community C. If home targets are inaccessible, the portal deals with it in the following ways:
  - My Pages
    - My Pages are not secured because users can only see their own My Pages. Therefore, a My Page home target is always accessible.
  - Communities
    - If the home community is not accessible, users default to the first community in their My Communities list.
    - If users are not members of any community, users default to the Join Communities page.
    - If users do not have access to any communities, they will not be able to access the portal at all.
  - Knowledge Directory
    - If the home folder is not accessible, users default to the top level folder.

## Community Best Practices

- If you have strict guidelines on the look and feel of communities, utilize the community and page templates since they will force how communities can look. Do not give users access to the Custom Page Template if you do not want community managers to control the overall layout of the community.
- If you force the use of page templates on communities, you can add and remove portlets from those communities by adding and removing portlets from those communities.
- If you do not want your community managers to make their own subcommunities or portlets, do not give them Select rights on any community, page, or portlet templates. Remember that community owners automatically have the right to add pages and create subcommunities, but they must have Select rights to community and page templates to do so. This is also true for portlets. Although community owners intrinsically have rights to create portlets from templates, they must have access to a portlet template to do so.
- You can use subcommunities as secured community pages. Since they are managed objects, they can have a different ACL than the parent community.
- If you want to have the same portlet multiple times in your community but with different preferences, you can use subcommunities as well. Since they are separate communities, they can have different community preferences.
- Utilize the difference in the ACLs to the community.
  - If you want somebody to manage portlets and the pages of the community but not change the memberships of the community, give that user Edit rights on the community.
  - If you want users to see a community but not join the community (allowing them to be guests of the community), give those users Read access.

## Incorporating Collaboration Server

How is Collaboration Server used in the Enterprise Web? As a foundation service, Collaboration Server often exists as an integral piece of any Enterprise Web Application. Applications with a focus on producing documents, transferring knowledge to a large external audience, or simply organizing a small team around a common goal often employ Collaboration Server as an application component. To illustrate some basic methods of deploying Collaboration Server, let us examine four common applications found in a number of Plumtree deployments.

### Sales Process Automation

Each account manager at a given organization manages several prospective clients. A sales process automation application powered by the Plumtree Enterprise Web Suite would enable account managers to create Web properties for each prospective client in their portfolio. Each of these Web applications would have a Collaboration Server project as part of the application template. The project itself would be based on a Collaboration Server project template that was region specific. For example, there might be two project templates total, one for United States sales and one for International. The template would then create a project with all of the appropriate legal contracts, product schedules, marketing documentation, and support contracts automatically preloaded.

Account managers would simply have to enter a new prospect's account name and e-mail address into the portal. The Web property would be automatically created and Collaboration Server would send a notification directly to the client. The account managers would also subscribe to the Collaboration Server project as a whole so that they would be notified when clients begin to interact. This online property would then be available to the prospective client from anywhere in the world. Clients could share documents, upload an RFP, assign various tasks to their account managers, post discussion questions on topics of concern, or schedule meeting events through the Collaboration Server project available in this application.

Not only would this application increase contact and communication between sales representatives and prospective clients, but it would serve to minimize unnecessary work load

on the sales representatives. Managing prospective clients becomes much easier with Collaboration Server notifications. Each time a client poses a question or uploads a new document, the sales representatives are instantly notified. In addition, because Collaboration Server is fully integrated into the Plumtree Corporate Portal, sales representatives can easily invite other employees with valid expertise into the Collaboration Server project.

## Customer Support

Most organizations today have some sort of support Web site. Collaboration Server can be used in this support context to improve response rate, decrease incident resolution time, and give customers an inside look to how their issues are being solved.

Assume that a customer can log in to an external support Web site and file incident reports for problems they are experiencing with a product. For each new incident that is filed, a Collaboration Server project is created and the relevant support engineer is notified. The engineer and the customer can then enter a discussion through the Discussions portion of the Collaboration Server project. Collaboration Server will keep track of all the messages that go back and forth and in the end will provide a clear record of the incident resolution. If this particular incident becomes common, the support engineer might forward the discussion information to the entire group so that everyone is immediately aware of the resolution.

The Collaboration Server project thus becomes the hub for every incident that Customer Support deals with. Each incident can be populated with the appropriate tasks and documents that are necessary to solve the customer's issue. The customer then has a real-time outlook into exactly what is being done to resolve the issue.

## Finance Application

At the end of each quarter, the accounting departments of most organizations run through a number of tasks and produce several documents detailing the organization's financial performance. Many Plumtree customers have created a Finance Dashboard which incorporates Collaboration Server to automate the process of closing their books.

Normally the heart of these applications is a Collaboration Server project template which outlines all of the necessary tasks, documents, and dates that must be met to close the



books. Each quarter a new project is created from this template and all of the necessary data is loaded up and all of the relevant employees are automatically assigned their tasks for the quarter. In addition, notifications are set up so that employees know when their particular tasks should start, when they are due, and which of their tasks are overdue. Employees come to the Collaboration Server project to examine their tasks, post questions or concerns, and check-in relevant documentation. The head of the finance department then has a consistent outlook onto the close process and is able to communicate results more quickly and accurately to the executive team.

Once the close process is complete, the project is archived and stored as a record. In the event of an audit, the finance department can easily pull up all of the actions that were performed each quarter by simply restoring old Collaboration Server projects.

## E-learning Application

A number of Plumtree customers in the Education field use Collaboration Server as an online classroom. Normally each class has an associated Collaboration Server project where the instructor posts a syllabus, assignments, and answers student questions. A similar application can be applied to E-learning in the corporate environment.

## How Not to Use Collaboration Server?

Although most organizations find Collaboration Server to be the most user-friendly method of sharing documents online, it is not meant to replace Enterprise Content Management (ECM) systems, such as Documentum or Stellent. Versioning features, such as roll-back, check-in/out, and version history are available in Collaboration Server, but the robust functionality normally associated with content management systems is not available through Collaboration Server. For example, there are no renditioning features or complex workflow systems currently available through Collaboration Server. However, all Plumtree products are designed to integrate and sit along side such enterprise applications and, in many ways, Collaboration Server provides complimentary functionality to ECM systems.

## Incorporating Content Server

Content Server allows IT personnel and community administrators to create a useful class of database-backed portlet Web applications. This class of portlet Web applications is characterized by the following elements:

- **Custom content types** - At the heart of many Web applications is one or more data types representing content that end-users will contribute, edit, and retrieve. For example, a “Departmental News” application requires the definition of a “News Article” content type with properties corresponding to headline, author, article body, department name, and so on. Similarly, an “Employee Book Reviews” application requires the definition of a “Book Review” content type with properties for reviewer, book title, synopsis, review body, reviewer rating, and so on. Content Server data entry templates allow the creation of database-backed custom content types without worrying about database programming or low-level machinery.
- **Custom presentation** - The presentation of content in a Web application should be separable from the details of specific content items. For example, in an “Employee Book Reviews” application, the look and branding of the individual reviews and a portlet entry point for accessing the latest reviews might need to be centrally managed by a community administrator, independently of the information about specific books contributed by the reviewers. Content Server presentation templates allow complete control over the HTML for display of summary views and individual content items.
- **Form-based submission** - While the setup and configuration of content-based Web applications is typically done by a small number of administrators, content is usually contributed by a much larger group of people. For example, in an “Employee Book Reviews” application, employees need an easy way to submit their reviews through a browser form. Over time, the application will contain a growing number of content items (the individual reviews) submitted by various people and presented in a uniform way. The Content Server Content Item Editor provides a rich browser form that contributors can use to submit content items and (if necessary) save and version their work.

With the building blocks of data entry templates, presentation templates, and the Content Item Editor, Content Server can be used to build a number of useful portlet Web applications for contributing content and surfacing it within the portal.

## The Content Server Portlet Templates

Content Server ships with a small number of portlet templates which are used to create specific portlets that can later be customized for a particular application. Customization involves determining properties for content types, designing presentation templates for individual types, and defining the presentation and functionality of the portlet view entry point for the application. Most such customization must be done through the Content Server Explorer, an administrative console that provides access to all Content Server administrative objects.

Content Server portlet templates include:

- **Announcement Portlet Template** - Encapsulates a simple pattern in which a rich text message will be displayed in a portlet. Powering this portlet is a single content type (“Announcement”) and associated presentation template and content item in which the rich text message is stored.
- **News Portlet Template** - Encapsulates a pattern in which there is a central content type which will be used to create individual items (“Articles”); a portlet view that acts as an entry point to create, search, and view the most recent items (“Main Page”); and a secondary view that aggregates information about all items of the given type (“Index Page”).
- **Community Directory Portlet Template** - Encapsulates a more complex pattern in which there are a number of central content types from which items can be created (three types of “Article”); a portlet view that organizes the items created from these types into a folder hierarchy (“Main Page”); and a number of secondary views that list all items in each folder (“Index Page” for each subfolder).
- **Web Page Portlet Template** - Allows the creation of a portlet associated with any pre-existing published content item. Unlike the other templates, this one does not create a new section and Content Server objects to be customized; rather it provides a way to display an existing content item in a portlet.

These templates, particularly the first three, provide a useful starting point for creating portlets to address specific business needs.

*Key Tip:* Only one portlet created from the templates can be associated with each Content Server section (the top-level folders which govern security and workflow processes for the Content Server objects they contain). This can sometimes lead to section sprawl, in which a broad, flat hierarchy makes the Content Server Explorer more difficult to use. It is important to note that the one-to-one association is not fundamental—it is possible in 5.0 to create a portlet web service with the URL of any published content item, and a portlet which uses the web service. This can be useful in allowing more than one portlet “window” into the items in a section. Note however that security and workflow are still managed on a per-section basis.

## Using Content Server to Create Workflow Applications

Whether you create Content Server-based portlets or use Content Server to manage pages in a non-portal Web site, you will often need to codify business processes that govern the movement of content from creation to review and editing to publication. Content Server provides an embedded Workflow Server which can be used to attach simple business processes to the content items managed in the system.



**Note:** Plumtree Workflow Server 5.0.x is installed as a service on the same machine as the Content Server Web application. The workflow engine is embedded in the sense that it does not require a separate application server container (it uses JBoss technology) and there is no separate user interface or access to the workflow engine.

There are a number of key concepts to understand in order to use Content Server to create applications with workflow:

- **Stages** - Sequenced activities, or stages, in which information workers can act on content items (edit and create new versions, review and approve) form the core of a business process.
- **Workflow Templates** - Also known as a process definition, a workflow template is a sequence of named stages describing a process at a high level. For example, a workflow

template for creating customer support knowledge base articles might consist of the following stages: Creation, Technical Review, Documentation Review, Final Editing. Similarly, a workflow template for purchase order approval might consist of Creation, Director Review, VP Approval, CFO Approval.

- “Attaching” Workflow - A workflow template can be attached to a Content Server section and filled in with specific user or group assignments for the stages. New content items created in the section can be automatically or manually submitted to the resulting workflow pipeline.
- Completing Workflow - A content item created in a section with attached workflow can be thought of as moving through the pipeline towards completion, typically resulting in publication. Items can later be resubmitted to workflow to go through the sequence of stages again.

A simple example of a Customer Profiles portlet can be used to illustrate the steps involved in creating a workflow-enabled Content Server portlet. (The *User Guide for Plumtree Content Server* provides technical details of setting up workflow.)

1. In the portal's administrative hierarchy, create a portlet starting from the News Portlet Template. This allows you to choose a name (for example, “Customer Profiles”) for the portlet and the Content Server section that will be created.
2. In Content Server Explorer (accessible through the Content Administration portlet), access the “Customer Profiles” section and customize the presentation templates for the portlet view (“Main Page”), aggregate view (“Index Page”), and individual profiles (“Articles”). Typically an HTML-savvy designer would contribute to this step by making high-gloss attractive presentation templates. The Main Page and Index Page must be published in order for the customizations to be visible in the portlet.
3. In Workflow Administration (accessible through the Content Administration portlet), create a Workflow Template for the desired approval process. For example, a process definition for customer profiles might have three stages with publication allowed only in the final stage: Creation, Profile Team Review, VP Marketing Approval.
4. In Content Server Explorer, attach the workflow template to the “Customer Profiles” section, and fill in assignments for each stage. Stages can be assigned to users or groups. For example, the Marketing Managers group could be assigned to the Profile

Team Review stage, and the actual Vice President could be assigned to the VP Marketing Approval stage.

After completing these steps, a community administrator or (if desired) an individual user can add the “Customer Profiles” portlet to a page, after which designated users can create new content items by filling out a form. When new items are submitted, e-mail notification goes to the Marketing Managers group, and anyone in that group can review, modify, and approve a profile. When a profile is approved at this stage, e-mail notification goes to the VP of Marketing, who can approve profiles for publication.

*Key tip:* If an administrator edits an attached workflow process definition, by default, the changes apply only to new content items, not to items that are already in process. In order to force pre-existing content items to follow the newly modified process definition, they must be detached from workflow using Content Server Explorer, and resubmitted to workflow.

*Key tip:* Only one workflow template can be attached to a section. If different workflow processes are needed for similar content, the content must be placed in different sections.

## Content Server Limitations

While Content Server can be a powerful tool for creating certain types of portlet Web applications, it has a handful of limitations that are important to understand in a deployment. Some of these limitations include:

- **Section-level security only** - Security settings can be applied to top-level Content Server sections only, not on a finer-grained subfolder or item level. As a result, if it is important to use security to isolate different types of content, multiple top-level sections are required.
- **Section-level workflow only** - A Workflow process can be applied to top-level Content Server sections only, not on a finer-grained subfolder or item level. So if it is important to have different content routed through different sequences of workflow stages, multiple top-level sections are required.
- **One Content Server portlet per section** - When you create a new Announcement, News, or Community Directory portlet from the corresponding template, a new top-level section is created in Content Server. It is not possible to create the portlet so

that its associated Content Server objects are created in a subfolder of an existing section. It is possible to move a top-level section into a subfolder of a different section, thus cleaning up the folder hierarchy and avoiding section sprawl. However, any security or workflow for the moved objects will be inherited from the section to which they are moved (according to the security and workflow limitations mentioned above).

## Similarities and Differences Between Content Server and Studio Server

Both Content Server and Studio Server can be used to create database-backed portlets with form-based data entry and aggregate portlet views. What are the differences, and when should you create a Content Server portlet versus a Studio Server portlet?

- If you need precise control over the presentation of individual content items (*records* in Studio Server parlance), you should create a Content Server portlet in order to take advantage of presentation templates.
- If you need to create multi-stage sequential workflows in which content items are routed to users and groups for approval before publication, you should create a Content Server portlet in order to take advantage of workflow templates and the underlying Workflow Server. As noted elsewhere in the Deployment Guide, it is possible to simulate simple workflows with Studio Server.
- If your content items are lightweight and resemble rows in an Excel spreadsheet or relational database (as opposed to Web pages), and if tabular viewing and editing of these rows is important, you should create a Studio Server portlet in order to take advantage of record browsing, lookup, and in-place editing functionality.
- If you need control over the layout of the data entry form to be used by content submitters, and if you have no other application requirement to the contrary, you should create a Studio Server portlet in order to take advantage of the WYSIWYG form layout editor. (All form-based data entry in Content Server takes place through the Content Item Editor, which can be annotated but not customized in a detailed way).

## Related Materials

- Chapter 2 of the *User Guide for Plumtree Content Server* gives a listing and brief overview of all the major objects and concepts in Content Server. It discusses security roles, Content Server objects such as data entry templates and content items, intrinsic administrative portlets, Content Server Explorer, which provides complete control over Content Server objects, and the various editors and interfaces used to alter presentation templates and other objects. Chapter 6 provides additional details on using Content Server Explorer.
- Chapters 7, 8, and 11 of the *User Guide for Plumtree Content Server* give technical details on creating and editing data entry templates (chapter 7), presentation templates (chapter 8), and individual content items (chapter 11). The editors and forms described can be accessed from Content Server Explorer or (if so configured) from the portlet and individual item views of Content Server portlets.
- Chapter 9 of the *User Guide for Plumtree Content Server* gives a brief description of the Content Server portlet templates.
- Chapter 10 of the *User Guide for Plumtree Content Server* gives technical details of Workflow, describing how administrators can attach workflow to sections and how content contributors can check in/check out/approve items to advance them through pre-defined workflows.



## Incorporating Studio Server

Studio Server allows users to create a wide variety of database-driven portlets for the Plumtree Corporate Portal. In understanding what types of portlets can be created with Studio Server, it is important to first understand the basic building blocks that make up a Studio Server portlet:

- All Studio Server portlets are associated with a database table used to store records. A Studio Server portlet will use one (and only one) database table. However, multiple Studio Server portlets may share the same database table.
- Reports are used to display records from the portlet's database table. Reports can be customized to filter records based on certain criteria, and present results in a particular order. Certain portlets can be configured with multiple reports, in order to allow data to be presented in a variety of ways. A special type of report, called a summary report can also be used to aggregate data from a given database table.
- Calendars are used to display event-based data.
- Entry forms are used to create and edit database table records. The layout and presentation of forms can be customized, and notification rules can be associated with forms to send e-mail to specific recipients when records are created, edited, or deleted through the form.
- Search forms are used to search for records in a Studio Server database table. Like entry forms, their layout and presentation can be customized.

With these simple building blocks, a surprising number of useful portlets can be created with Studio Server to collect, organize, and share data within a workgroup.

## The Studio Server Frameworks/Templates

When creating a new Studio Server portlet, users start from a specific Studio Server portlet template or framework. A framework can be thought of as the basic outline of the user interface elements that a portlet contains, how those elements are associated with each other, and how end-users will ultimately interact with the portlet. Customizing the portlet in the Studio Server Wizards involves filling in the details: creating the database and cus-

tomizing things like entry forms, reports, and so on. For example, the Data Submission framework contains an entry form for users to submit records as well as a report that allows them to view records that they have created. Using this framework, users can create portlets, like work order requests, by setting up a database with columns for request description, priority, due date, and so on.

Studio Server includes the following frameworks/templates:

1. **Record Browser**, which allows users to view records in reports and create and edit records.
2. **Record Lookup**, which allows users to search for records in a database.
3. **Record Summary**, which displays aggregated data from a database.
4. **Data Submission**, which allows users to submit records to a database and view records that they have created.
5. **Poll**, which allows users to “vote” on some topic and see a summary of the poll results.
6. **Survey**, which allows users to answer questions by filling out a form.
7. **Calendar**, which displays event-based data.

## Sharing Databases Across Studio Server Portlets

One precept of effective portlet design is that each portlet should be designed to accomplish a specific objective for a specific set of users. In applying this rule to Studio Server portlets, it is important to recognize that while certain portlets allow you to accomplish many objectives (the Record Browser framework, for example, allows you to create, edit, browse, and search for records), it may often be more desirable to create a number of single-purpose portlets to accomplish each objective individually. You can do this by sharing a database table among several Studio Server portlets.

Consider the following example. You want to create a simple work order system to allow employees to request things like office supplies. Users of the system would include the employees of your company (who request supplies), an operations clerk (who places orders), and a receiving clerk (who processes incoming shipments and delivers the supplies). One strategy might be to create one Studio Server record browser portlet for all

these users. It could be confusing to employees, however, to create work orders from the same portlet that the operations clerk uses to track orders that need to be processed.

A better solution in this case would be to create three portlets that share the same Studio Server database table. A data submission portlet could be used by employees to submit work order requests; a record browser portlet, with reports to display pending and submitted orders, could be used by the operations clerk to track requests and orders; and a record lookup portlet could be used by the receiving clerk to find work orders by querying for a purchase order number. With all three Studio Server portlets sharing the same database table, each set of users can have a portlet custom tailored to their needs.

## Using Studio Server to Create Workflow Applications

Unlike Plumtree Content Server, Studio Server does not have mechanism for creating workflow templates that can be associated with a given database table or portlet. It is, however, possible to create portlets in Studio Server that simulate rudimentary workflows using the basic building blocks that Studio Server provides. The approach, in general, is to add various status and assignment fields to a Studio Server database table, and then to create various portlets and reports that filter records based on the values of these fields. E-mail notification can also be used to alert people when records are created or modified, thus alerting them to items that may require their attention.

For example, imagine that you wanted to add workflow capability to the work order request system described in the previous example. Before having the operations clerk place the order for supplies, you want to first have the employee's manager approve or reject the request. Only requests that were approved would be subsequently processed. In this case, you would add two fields to the database table used by the portlets in the system - "Manager to Approve" (Portal User data type) and "Work Order Status" (Text data type with a value list containing "New"—the default value, "Manager Approved", "Manager Rejected", "Order Placed", and "Order Delivered"). In the data submission portlet used by employees to submit requests, the "Manager to Approve" field would appear on the request form, while the "Work Order Status" field would be hidden. We would also want to create a notification rule for the form, so that an e-mail gets sent to "Manager to Approve" so that they would be alerted to a new order requiring their attention. Next, we would create a new record browser portlet called "Work Orders to Approve" with a report that listed records

where the “Work Order Status” field value was equal to “New” and the “Manager to Approve” field was equal to the value of the currently logged in user. We might also specify permission on the portlet so that it is visible only to managers in the company. When a manager viewed this portlet from a My Page or community page, it would list all the work orders with approval pending. They could then open these records and change the status to either “Manager Approved” or “Manager Rejected” as necessary. Similarly, we could also change the record browser portlet used by the operations clerk who places the orders to only display those records where the status was “Manager Approved.”

In this way, using e-mail notifications and filters, in conjunction with assignment and status fields, allows us to simulate simple workflows with Studio Server.

## Accessing Data in the Studio Server Database Tables

Under the hood, each Studio Server database table maps to a real table in the SQL Server or Oracle database configured for Studio Server. Using tools like Enterprise Manager you can easily see how these tables are defined. This is useful to understand, because it means that it is possible, assuming you have a basic understanding of how to write SQL queries, to access the data in Studio Server portlets from other, third-party applications. For example, you can use report writers to create more complex reports than what is possible in Studio Server itself. You might even have other custom written portlets that use data from various Studio Server tables.

Using ODBC, you can even access data from Studio Server portlets through Microsoft Office applications like Excel. Excel allows you to define SQL queries against an ODBC data source, and have the query results formatted and displayed in a spreadsheet. In the case of the work order example, employees in the accounts payable department might want to create an Excel spreadsheet listing all processed orders. While it is possible to export data from Studio Server portlets directly to an Excel spreadsheet, sometimes it might be more elegant to simply pull the data into the spreadsheet. The accounts payable employee might create a spreadsheet with a query pointing to the Studio Server database table, and then use a VB macro to execute the query, retrieve the results, and format them as desired.

One caveat to accessing data in the Studio Server database tables is that you should only do this to read data, not to create or modify records. The reason for this is that Studio Server makes extensive use of caching in order to improve performance. If another application or

process were to modify the data in a Studio Server database table, Studio Server would not know that it had changed and would still display the older (cached) version of the record.

## Studio Server Limitations

While Studio Server makes it possible to easily create a wide variety of portlets for the portal, it does have limitations which preclude it from being a good tool for certain types of portlets you might want to create. Some of these limitations include:

- The user interface in Studio Server portlets cannot be customized beyond what you can do with in the Studio Server portlet wizards. For example, you cannot customize the actual HTML that is used to display a form or report. If you need precise control over the user interface and formatting of your portlet, you might instead consider using Plumtree Content Server or creating a custom portlet.
- Studio Server does not offer a robust workflow process modeling facility. While there are ways, as described above, of using Studio Server to handle simple workflows, there is no way, for example, to define things like routing rules, conditional branching, and so on that you would find in a workflow or process automation tool. If you need more robust workflow capability, you might instead consider using Plumtree Content Server or creating a custom portlet.
- Studio Server does not allow you to create portlets that use more than one database table, as in a relational database system. It is not possible, for example, to create data modeled in one-to-one, one-to-many, or many-to-many relationships. If you need to create portlets with data structured in such ways, you should use a custom designed portlet.
- Studio Server portlets can not be configured to use non-Studio Server database tables, such as those used by existing business applications. If you need this capability, you should use a custom designed portlet.

## Incorporating Analytics Server and Analytics Server Portlets

Plumtree Analytics reports are intended to make usage metrics visible to a limited set of administrative users who perform particular business functions, such as capacity planning,

QoS analysis, ROI analysis, “best bet” customization for Search, and the like. Plumtree Analytics is not intended to be a business intelligence tool.

The Plumtree Analytics Console and portlet reports contain usage data that is valuable for enterprise portal analysis but might be regarded as private or sensitive to portal users. For example, Search, Document, Community, and Portlet reports can be configured to display activity metrics for a particular user, based on several user properties, such as Email Address, First Name, or Last Name.

To protect security and privacy interests before you roll out Analytics Server reports:

1. Manage administrative access to Plumtree Analytics Console and portlet templates.

To ensure only a limited number of administrative users can add the Analytics Console community to their My Communities or create portlets based on the Plumtree Analytics Portlet Templates, create a new administrative group and manage group membership accordingly. Members of this administrative group require Read access to the Analytics Console community and Select access to the Admin Objects directory that contains the portlet templates.

2. Manage user access to Plumtree Analytics portlets.

When you create portlets, configure metrics that do not contain private or sensitive data unless such a view is particularly intended. If the metrics in the report do contain private or sensitive data, configure security so that only appropriate, specified users have Select access and can therefore add the portlet to their My Pages.

3. Ensure that guest users are never allowed to add Analytics portlets to their My Pages.

For information on creating portlets from portlet templates, see the *Administrator Guide for Plumtree Corporate Portal*.

For information on creating administrative groups and managing user and guest access, see the *Administrator Guide for Plumtree Corporate Portal*.



**Note:** Users should not add many Analytics portlets to a single My Page. The more portlets you add to a My Page, the slower the performance. If users experience-

unacceptable performance or timeouts, you can recommend that they include fewer Analytics portlets on each My Page.

## Incorporating Branding (Customizing the User Interface)

The Plumtree Corporate Portal and its related server products support a wide array of user interface customization techniques, most of which do not require special programming skills. Before we list guidelines for determining the best solution for you, we will review the possibilities:

### Subportals

Subportals define many aspects of the user interface for broad groups of users. Subportals control what your start page is when you log in, the features available to you, what your navigation looks like, and what mandatory links are shown in your navigation. Which subportal a user sees is determined by what folder that user is stored in.

Users are organized into administrative folders. These folders, in turn, are populated and maintained by a user synchronization process that retrieves user, group, and profile information from a remote authentication source such as Active Directory or LDAP. The administrative folder hierarchy normally reflects the hierarchy within a company. For example, you might have folders named Marketing, Sales, Product Management, Support, and IT, and each of them could have a different set of features and navigation, determined by the subportal they see.

For more information on subportals, refer to the *Administrator Guide for the Plumtree Corporate Portal*.

### Navigation

Portal navigation is customizable, in fact, the portal ships with seven different navigation types out of the box. Navigation controls everything outside the center of the page, not

including the header and footer. To change the navigation presented to a user, edit the “Navigation Options” in the subportal editor.

Navigations are “pluggable”, that is, you can develop new navigations using programming languages like C#, Visual Basic .NET, or Java and add those navigations to your portal.

Although navigations are associated with subportals, each navigation can be very dynamic, displaying a completely different look for each page type, each user, or any other settings you like. For example, Plumtree's Support Center navigation shows completely different HTML when you are on a Support Center community, even though it is all done within a single pluggable navigation view. The Support Center navigation is downloadable from the Developer Center.

## Style Sheets and Portlets

If you are happy with the layout of your existing portal and simply want to change things such as fonts, colors, logos, and images, you can override all those settings by changing your style sheet. Since we support localization of our style sheets into many languages, the easiest way to modify a style sheet for a multi-language portal is to use the Style Sheet Mill, which takes values from template files and uses them to generate style sheets in multiple languages using localized text from our translation files. You can find out more about the Style Sheet Mill in the *UI Customization Guide*.

## Branding

Perhaps you want to change the look in a dramatic way for each community in your portal, but you do not want to do any programming. For that, you would use Plumtree Content Server to create a branding header portlet or branding footer portlet. Content Server lets you create a variety of different portlet types and manage their content through a Web interface, making HTML, image, and style sheet changes very easy. The Content Server documentation includes several examples, and there are downloadable branding examples on the Plumtree Support Center as well. By assigning individual branding header or footer portlets to each subportal and each community, you can dramatically change the look for both groups of users and individual communities without writing code, and you can manage the content for complex deployments remotely.



## Pluggable Event Interfaces (PEIs)

Sometimes you want to add new functionality rather than modify existing functionality. Plumtree has implemented a large number event categories you can hook into, each with several different event types. For example, you might change the behavior after a user logs in; for users who had not yet filled in their user profiles, you could have them redirected to the user profile form. To accomplish this, you would need to implement not only a PEI, but a custom activity space, model, view, and controller for any special landing pages you wanted to write from scratch. You could also use Dynamic Discovery to override a view class for an existing page.

## Custom Activity Spaces

You might want to precisely control the exact look of the center of the page, as well. For example, you might want to control how portlets are rendered on the page. The file `MyPortalContentView` renders portlets into columns based on your page layout style. You might want to redesign that page center so the portlets are arranged in rows instead of columns. For this, you would need to override the default view with your own, using dynamic discovery, also outlined in the *UI Customization Guide*.

A more forward-compatible approach involves extending `ActivitySpaces`, by creating new `ActivitySpaces` and views that extend existing ones and directing PEIs and other links to those new spaces. Then, as Plumtree improves existing activity space components, your code will benefit.

## Putting It All Together

Here are some simple rules to help you determine which of these technologies is appropriate to your needs.

- The portal has to precisely match another Web site's look.

If you want pixel-perfect alignment with another Web site's look, then writing a custom pluggable navigation is your best option.

If all you need to do is match colors, fonts, logos, and the like, you will find it faster to use the Style Sheet Mill and specify new values for standard style elements.

If you want to change all the colors, fonts, and logos, but want to preserve flexibility for future design changes, your best bet would be to use Content Server to create one or more branding header and footer portlets to use on your default subportal or key communities.

Examples of pluggable navigations can be downloaded from the Plumtree Support Center. We also ship the source code for all seven of the navigations available in the portal.

- The portal requires a completely different and unique layout and design.

You might need to look at going beyond a custom pluggable navigation and override several of the core view classes used to render the center of each page type. Although this requires more programming, you can base your new views off of the existing code, which we ship with the product.

- You need the portal to behave differently when a user completes a specific action.

You probably need to write a PEI. Keep in mind that PEIs are not just for redirecting to another page; they can be used to record information to a database, to check or verify information gathered from the server, to store a flag or special note in the session, or otherwise respond to an event. The PEI interfaces and all their related events are listed in the *UI Customization Guide*; there are over fifty different events you can subscribe to, associated with before, after, or when an action occurs.

- You want one group of users to have a different experience from another

That is exactly what subportals are for. The most common use of a subportal is to assign all employees to a company subportal that has all features enabled and uses a regular navigation, while the default subportal uses a null or empty navigation that only shows a restricted set of links for external users (that is, all users who are not part of the company subportal). You might find that the regular navigations shipped with the portal, when combined with Content Server branding, give you all the flexibility you need.

## Common Requests and Solutions

- I want to switch a user's subportal as they navigate.

The server API (see the Plumtree Support Center) does allow you to change which subportal is associated with a user, but this a very complex operation and performance would be slow. The subportal was never intended to be used this way.

A better question to ask is: What do you need to change? If all you are looking to change is the look plus the links the user sees, no matter how radical, a pluggable navigation is a much better way to go and performance is excellent. Navigations do not have to be fixed; the HTML displayed, the links used, the colors, fonts, javascript, and/or Flash animations shown can all be dynamically driven off of any of a number of session variables, user profile settings, server API calls, extended object data properties, or values loaded from an XML file.

- I want to be able to edit the HTML of a portlet and change the style sheet of a portal.

Use a Content Server-driven branding header portlet. With a few clicks, you are in an HTML editor that includes a tag helper than supports dynamic values. You can upload new images and style sheets, and changing from one style sheet to another is as simple as selecting from a list.

You might also want to look at using a content canvas portlet. These portlets displays across the top of a community page and, when combined with content server, are an excellent way to keep employees informed and up-to-date on latest events and news.

- I want to use one portal for two domain addresses, but have a different look for each domain.

You can use a pluggable navigation that checks the referrer URL and displays a different look for each domain.

You might also want to override, with dynamic discovery, the login page to allow you to change the look of login based on the domain.

You might consider implementing a PEI if you want to have the guest user's default home page be different based on the domain; using an OnAfterLogin PEI, you could

redirect the guest user to different communities based on the domain in the request header.

- I do not want to see a login page at all.

If you want to skip the login page entirely for the Guest user but do not want to write a PEI, there is a setting in the portal config file (**j\_config.xml** or **n\_config.xml**) under your the **\settings\config** directory that you must change. The property is under the Authentication section and is called `GuestRedirectLogin`; if you set the value to 0, you will automatically be redirected to the Guest user's My Page.

You can then use a login portlet on that page to allow employees to log in. You should set up a separate subportal for employees to control their features and navigation options.

- I have a complex authentication scheme and I do not want to see the authentication source drop-down list on the login page.

You might want to consider writing a custom authentication source. You could then authenticate against other existing authentication sources in a custom way and configure your portal to use only your custom authentication source. When there is only one choice possible, the authentication source drop-down does not display.

Alternatively, you could write an `OnBeforeLogin` PEI that does customized authentication directly in the portal. This is not considered the most scalable or performant approach, but it is simple.

- Can I write Navigations, PEIs, and Activity Spaces in Visual Basic .NET?

Yes. We include a sample PEI written in Visual Basic in the sample code that ships with the portal. If you are running the .NET portal, any language supported by Microsoft's CLR will work, including Visual Basic .NET, C#, and C++.

- How do I add links to a portlet that refer to other pages in the portal?

You should download and review the Enterprise Web Developer Kit (EDK) and look up the `pt:opener` tag. For ease of use, you can use the Tag Helper in the Plumtree Content Server's branding portlet wizard. There is a downloadable quick start on branding available in the Plumtree Support Center that makes this easy to find.

- How do I turn off the portal banner (where the login and search fields are)?

This is a pluggable navigation feature. Open the source code for any of the navigations we ship with the portal and look at the `IsFeatureEnabled()` method of any of the `INavType` classes. You can disable the portal banner view, as well as the header and footer portlets.

- I would like to have different style classes assigned to each column of portlets in the page.

You will need to completely override the `MyPortalContentView` and modify the `HTMLElements` to use the new style classes. The existing portal code uses the same styles for all column elements.

- I need to make a portlet that includes links to other community pages.

We already wrote it for you. It is called the Community Knowledge Directory portlet and can be created from the Community Knowledge Directory. Read more about it in Plumtree Knowledge Base article: DA\_197368 “Creating Community Content Snapshot Portlets for Navigation.”

- I want to add links to my navigation that go to pages outside the portal.

Just add them as links to your subportal. You will find this feature in the Subportal Editor. They will show up on a mandatory links tab in the navigation of all users of that subportal. You can add Web links, links to Knowledge Directory folders, links to documents, and links to communities or community pages.

- I want to change the colors and styles of my portal without using the Content Server.

Use the Style Sheet Mill as outlined in the *UI Customization Guide*, or try the Style Sheet quick start on the Plumtree Support Center. This will allow you to modify the existing styles and regenerate them for all languages or write a new style sheet and add it to the mill. You can also write a header portlet that uses the EDK to set an administrative preference called 'Portal-Style' to the URL of your new style sheet.

- I would like to change the text messages used throughout the portal.

These are in the message files installed with your portal, under **settings/i18n**. There are several files and each have been translated into multiple languages. We update these

with every major and minor release of the portal, so be sure to back up your changes so they can be merged in. We are careful to only add new strings rather than remove old strings in order to stay backwards-compatible.

If you need to add new strings, add them in a new file, such as **SampleMsgs.xml**, and load them separately in your code. Then you can be sure that the strings you added will not conflict with strings added by Plumtree in the future.

- I would like to create my own top bar view.

You should override the top bar view with your own implementation using dynamic discovery, or write a pluggable navigation that turns off the top bar and add what you need to your header portlet.

- How do I suppress links to “Related Communities” and “Subcommunities”?

You will need to write a new pluggable navigation that simply does not display links to those features. The pluggable navigation used on our Support Center is downloadable sample code from the Support Center, and it demonstrates this.

- How do I get information about the page I am going to in the OnBeforeLogin PEI?

Because you have not yet logged in, it is not possible to determine in advance where your subportal would send you; it is also possible there is already an OnAfterLogin PEI that would redirect you somewhere else. In OnAfterLogin, however, you have access to the current page ID, community ID (if you are on a community) and user and session information. You can then redirect somewhere else entirely based on that data.

- I just want to change the greeting text in the portal banner and login page.

If you want to make the greeting dynamic, you need to override and replace the portal banner view and/or the login page view.

The guest user's greeting is stored in one of the message files, under **settings/i18n/<language>/ptmsgs\_portalcommonmsgs.xml**, entry #315.

Each user has an individual configurable greeting they can change in their user settings. The default value of the greeting (“Welcome, “) can be found in **ptmsgs\_portalcommonmsgs.xml**, entry #546.

- Is it possible to use ASP or JSP pages in the portal?

Yes. You can specify ASP or JSP pages as links or as destinations of Redirect objects. ActivitySpaces and PEIs can therefore redirect to them in the portal. You can use the EDK calls and server code within ASP and JSP pages, allowing you to access many of the features and functionality of the portal, although this approach should be used more as an expedient than a proper design.

ASP and JSP pages cannot be used within an ActivitySpace or a pluggable navigation.

## Rolling Out the Enterprise Web

After planning and creating your Enterprise Web, you must decide how you want to market your Enterprise Web so you can be sure your users will take advantage of it. Here are a few marketing ideas:

- Broadcasts and giveaways
  - E-mail teasers
  - Voice mail commercials
  - Portal mouse pads
  - T-shirts with portal logo and URL
  - Portal screen saver
- Newsletters
  - Executive endorsements
  - Screen shots
  - Features
  - Q&A
  - Glossary of portal terms
  - Team contacts
  - Training dates
- Posters and banners
  - Post ads for HR community in place of wall boards

When you first deploy the Enterprise Web to your users, you should introduce them to the features. For example, you might consider the following:

- Flash tours
  - Lead users from intranet to portal with a feature tour
  - Link to the portal directly from the tour



- Online tours
  - Clients and prospects can preview portal services
  - Cost-effective
  - Convenient
- Video tours
  - Testimonials from business leaders drive adoption
  - Educational and easy-to-watch

Probably the most important thing you can do to make sure your users will continue to use your Enterprise Web is to put help within reach:

- Online:
  - One portlet answers most users' questions; place prominently on default My Page
- Offline:
  - Concise portal handbooks issued to every employee
  - Annotated portal screen shots
  - FAQs
  - Key contacts

You might also consider adding some fun to your Enterprise Web so that users *want* to log in every morning and so they do not have to go anywhere for a break:

- Portlet games and quizzes
  - Pique interest in your portal
  - Users stay at their desks to relieve stress
- Keep users coming for daily laughs or lessons
  - Cartoon strips
  - Quote of the day
  - Tip of the day
  - Magic 8-ball fortune teller
  - Horoscopes



# 5

## Production Maintenance

This chapter is written for Enterprise Web administrators and/or IT personnel tasked with maintaining the Enterprise Web. It describes how to maintain the Enterprise Web, troubleshoot problems, and monitor performance.

### Production Maintenance Checklist

Here are a few suggestions for periodic tasks you should consider to maintain your production system:

- Daily
  - Modify security of portlets, communities, and other objects in the portal
  - Modify permission roles for users
  - Publish new and existing applications/portlets to Remote Servers
  - Monitor Portal Server, Database, and Remote Server alerts for CPU, memory, and hard disk usage to ensure availability
- Weekly
  - Install releases to one or more Enterprise Web servers
- Monthly
  - Add new hardware to the environment (for example, new Remote Server, new hard disk, and so on)
- Ad Hoc
  - Install portal patches
  - Install server patches from Microsoft/Dell/Antivirus

## Troubleshooting Tools

There are several tools available to help determine the source of problems with your portal. Plumtree provides PTSpy while other tools already exist on your machine. These tools can help solve portal problems.

Read this section to familiarize yourself with the following troubleshooting tools:

- [“Performance Monitor Counters” on page 5-2](#)
- [“Plumtree Analytics Server” on page 5-13](#)
- [“PTSpy” on page 5-17](#)
- [“View Source” on page 5-22](#)
- [“ODBC Testing” on page 5-24](#)

### Performance Monitor Counters

The Plumtree Corporate Portal includes several performance monitor counters to help you optimize your portal performance. The Performance Monitor performs the following functions:

- Charts real time performance data
- Logs performance data
- Alerts conditionally on performance data
- Reports current value of performance data

For more information on the Performance Monitor application, refer to the Performance Monitor documentation and online help.

## Performance Monitoring Uses

You might use the Plumtree performance counters to:

- chart or log total usage counts for different user activities.
- log Failed Logins for security audits.
- find widespread problems by monitoring Errors.
- test and troubleshoot your portal.

To test and troubleshoot your portal, you should use a combination of PPE, Database, Authentication, and PortalPages counters. These counters can help you identify patterns preceding an instability in the portal, investigate different causes of problems. During testing, these counters can also help you find bottlenecks in your Plumtree system and measure performance and load on your Plumtree Servers.

Use the Performance Monitor alerts to:

- run a program when particular conditions occur.
- alert you when CPU usage gets too high.
- alert you when Memory availability gets too low.
- alert you when ASP Requests Queued gets too high.
- alert you when the PPE goes down and automatically reboot.

Use Performance Monitor counters to provide a trend analysis of resources:

- Log performance data over long periods.
- Determine when new hardware will be required.
- Determine what bottlenecks exist.
- Determine what tuning would be most beneficial.

Available Performance Monitor Counters

The following Plumtree performance monitor objects (groups) and their counters are available:

Table 5-1: Available Plumtree performance monitor counters  
(Sheet 1 of 5)

Object	Counters	
Plumtree: Authentication	<ul style="list-style-type: none"><li>• Authentication Concurrent Calls</li><li>• Authentication Queries/sec</li></ul>	<ul style="list-style-type: none"><li>• Authentication Query Time</li><li>• Total Authentication Queries</li></ul>
Plumtree: Binary Gateway	<ul style="list-style-type: none"><li>• Bytes received</li><li>• Bytes received/sec</li><li>• Bytes sent</li><li>• Bytes sent/sec</li></ul>	<ul style="list-style-type: none"><li>• Download Requests</li><li>• Download Requests/sec</li><li>• Upload Requests</li><li>• Upload Requests/sec</li></ul>
Plumtree: BSTR Allocs	<ul style="list-style-type: none"><li>• Allocated BSTRs</li><li>• BSTR Allocs/sec</li><li>• Total BSTR Allocs</li></ul>	<ul style="list-style-type: none"><li>• Total BSTR Frees</li><li>• Total BSTR Reallocs</li></ul>
Plumtree: Database	<ul style="list-style-type: none"><li>• Database Concurrent Calls</li><li>• Database Queris/sec</li></ul>	<ul style="list-style-type: none"><li>• Database Query Time</li><li>• Database Query Totals</li></ul>

*Table 5-1: Available Plumtree performance monitor counters  
(Sheet 2 of 5)*

Object	Counters
Plumtree: Intrinsic Portlet Provider	<ul style="list-style-type: none"><li>• CommunityPublications Portlet Generation Time (milliseconds)</li><li>• CommunityPublications Portlet Requests/sec</li><li>• Java Portlet Generation Time (milliseconds)</li><li>• Java Portlets Requests/sec</li><li>• Login Portlet Generation Time (milliseconds)</li><li>• Login Portlet Requests/sec</li><li>• MyFolders Portlet Generation Time (milliseconds)</li><li>• MyFolders Portlet Requests/sec</li><li>• MyPublications Portlet Generation Time (milliseconds)</li><li>• MyPublications Portlet Requests/sec</li><li>• Search Portlet Generation Time (milliseconds)</li><li>• Search Portlet Requests/sec</li><li>• Total CommunityPublications Portlet Requests</li><li>• Total Java Portlets Requests</li><li>• Total Login Portlet Requests</li><li>• Total MyFolders Portlet Requests</li><li>• Total MyPublications Portlet Requests</li><li>• Total Search Portlet Requests</li></ul>

Table 5-1: Available Plumtree performance monitor counters  
(Sheet 3 of 5)

Object	Counters
Plumtree: MPPE	<ul style="list-style-type: none"><li>• Avg Request Size</li><li>• Bytes Received/sec</li><li>• Bytes Sent/sec</li><li>• Client Errors/sec</li><li>• Connect Delay</li><li>• DNS Time</li><li>• Net Reliability</li><li>• Receive Delay</li><li>• Requests Sent/sec</li><li>• Resolver Delay</li><li>• Response Size</li><li>• RTT</li><li>• Send Delay</li><li>• Server Delay</li><li>• Server Errors/sec</li><li>• Server Information replies/sec</li><li>• Server Is Up</li><li>• Server Redirects/sec</li><li>• Server Retries/sec</li><li>• Server Successes/sec</li><li>• Socket Closures/request</li><li>• Socket Errors/request</li><li>• Socket Errors/sec</li><li>• Socket Reuse</li><li>• Sockets Closed By Server/sec</li><li>• Sockets Closed/sec</li><li>• Sockets Connected</li><li>• Sockets Opened/sec</li><li>• Sockets/request</li><li>• SSL Certificate Errors/sec</li><li>• SSL Connections Closed/sec</li><li>• SSL Connections/sec</li><li>• SSL Contexts</li><li>• SSL Errors/sec</li><li>• Total Bytes Received</li><li>• Total Bytes Sent</li><li>• Total Client Errors</li><li>• Total Requests Sent</li><li>• Total Server Errors</li><li>• Total Server Information replies</li><li>• Total Server Redirects</li><li>• Total Server Retries</li><li>• Total Server Successes</li><li>• Total Server-Initiated Socket Closures</li><li>• Total Socket Closures</li><li>• Total Socket Errors</li><li>• Total Sockets Opened</li><li>• Total SSL Certificate Errors</li><li>• Total SSL Closures</li><li>• Total SSL Connections</li><li>• Total SSL Errors</li></ul>



*Table 5-1: Available Plumtree performance monitor counters  
(Sheet 4 of 5)*

<b>Object</b>	<b>Counters</b>
Plumtree: Object Counts	<ul style="list-style-type: none"> <li>• Current Objects</li> <li>• Number of constructions</li> <li>• Number of destructions</li> </ul>
Plumtree: Object Sizes	<ul style="list-style-type: none"> <li>• Current size</li> <li>• Total allocated</li> <li>• Total freed</li> </ul>
Plumtree: PortalPages	<ul style="list-style-type: none"> <li>• Admin Searches</li> <li>• Admin Searches/sec</li> <li>• Advanced Searches</li> <li>• Advanced Searches/sec</li> <li>• Banner Searches</li> <li>• Banner Searches/sec</li> <li>• Community Pages Viewed</li> <li>• Community Pages Viewed/sec</li> <li>• Document Click-throughs</li> <li>• Document Click-throughs/sec</li> <li>• Errors</li> <li>• Errors/sec</li> <li>• Failed Logins</li> <li>• Failed Logins/sec</li> <li>• Gateway Pages Viewed</li> <li>• Gateway Pages Viewed/sec</li> <li>• MyPages Viewed</li> <li>• MyPages Viewed/sec</li> <li>• Selection Searches</li> <li>• Selection Searches/sec</li> <li>• Successful Logins</li> <li>• Successful Logins/sec</li> <li>• Total Hits</li> <li>• Total Hits/sec</li> <li>• Total Open Sessions</li> <li>• Total Open Sessions/sec</li> </ul>
Plumtree: Portlet	<ul style="list-style-type: none"> <li>• CPU Cycles</li> <li>• Latency ms</li> <li>• Requests Executing</li> <li>• Requests per sec.</li> <li>• System time <math>\mu</math>s</li> <li>• Total Errors</li> <li>• Total Requests</li> <li>• User time <math>\mu</math>s</li> </ul>

Table 5-1: Available Plumtree performance monitor counters  
(Sheet 5 of 5)

Object	Counters	
Plumtree: Portlet Cache	<ul style="list-style-type: none"><li>• Cache Validations/sec</li><li>• Cached content size</li><li>• Errors Masked/sec</li><li>• Fresh Cache Hits/sec</li><li>• Hit rate</li><li>• Portlet Requests/sec</li></ul>	<ul style="list-style-type: none"><li>• Total Cache Validations</li><li>• Total Errors Masked</li><li>• Total Fresh Cache Hits</li><li>• Total Portal Requests</li><li>• Total Uncacheable Responses</li><li>• Uncacheable Responses/sec</li></ul>
Plumtree: Unified Cache	<ul style="list-style-type: none"><li>• Cache Hits</li><li>• Cache size</li><li>• Deletes</li><li>• Deletes/sec</li><li>• Hash table utilization</li><li>• Hit rate</li><li>• Hits/sec</li><li>• Inserts</li><li>• Inserts/sec</li></ul>	<ul style="list-style-type: none"><li>• LRU calls</li><li>• Max Objects/Hash entry</li><li>• Objects in cache</li><li>• Objects/Hash entry</li><li>• Searches</li><li>• Searches/sec</li><li>• TTL calls</li><li>• Utilization %</li></ul>

## Running Performance Monitor Counters



**Note:** You must have set up Profile System Performance correctly, and you must run the Performance Logs and Alerts service as a user with appropriate rights.

To use the Plumtree performance monitor counters:

1. Click **Start | Programs | Administrative Tools | Performance**.
2. In the performance monitor application, right-click the graph and click **Add Counters**. This displays a dialog box that allows you to add counters to your view:
  - a. Select the computer you want to monitor.
  - b. Select the performance object that includes the counter you want to add.
  - c. Select one or more performance counters to add to your view.
  - d. If appropriate, select one or more instances to monitor.
  - e. When you are finished, click **Add**. The specified counters are added to your view.

To save counter results in logs:

1. In the Performance monitor application, expand **Performance Logs and Alerts**.
2. Right-click **Counter Logs** and select **New Log Settings**.
3. Type a name for your log and click **OK**.
4. In the dialog box, add the objects and counters you want to log, specify how frequently you want to log the data, and click **OK**.

## Performance Monitoring Suggestions

This section lists the suggested Performance Monitor counters you should consider watching to ensure the health of your Enterprise Web.



**Note:** You can display information about a Performance Monitor counters by clicking the “Explain” button in most Performance Monitor Counter user interfaces. For example, here is the information, for ASP.NET's “Requests Current” Counter:

The current number of requests, including those that are queued, currently executing, or waiting to be written to the client. Under the ASP.NET process model, when this counter exceeds the requestQueueLimit defined in the processModel configuration section, ASP.NET will begin rejecting requests.

For detailed information about any of the counters discussed in this section, display the explanations.

### *General System Health (All Servers)*

- Processor
  - % Processor Time (\_Total)  
Provides basic CPU utilization for the total of all CPUs.
  - % Processor Time (each CPU)  
Provides basic CPU utilization for each CPU.
- System
  - Context Switches/sec  
Context Switches/sec is a very important CPU usage statistic. When a system becomes heavily stressed, Context Switches/sec will grow to 10,000 or more. In some cases, on multiprocessor systems, such stress will overload a system, even if CPU use remains relatively low.
  - Processor Queue Length

- Process (for each Server Process of interest)
  - % Handle Count
  - % Processor Time
  - Private Bytes

The amount of allocated memory for the process, including physical memory (RAM) and memory paged to disk.
  - Thread Count
  - Virtual Bytes

The amount of Address Space reserved for memory allocation to this process, including all memory already allocated.
  - Working Set

The amount of physical memory used by the process.
- Paging File
  - % Usage (\_Total)
- Physical Disk
  - % Disk Time
  - Disk Transfers/sec

Large upward changes in either of these indicate heavy disk activity.
- Network Interface
  - Bytes Total/sec

Network saturation can be the cause of some performance issues. This counter gives basic indication of bandwidth use.
- Memory
  - Available MB

The amount of physical memory free for processes to use.

### *Portal Server Specific*

- Plumtree: Portal Pages
  - All Page per sec counters
  - Total Hits/sec
  - Total Open Sessions
- Plumtree: Portlet Cache
  - Portlet Requests/sec
  - Fresh Cache Hits/sec
  - Cache Validations/sec
  - Errors Masked/sec
  - Hit Rate
  - Cached Content Size

### *.NET Specific*

- .NET CLR Memory
  - # Bytes in all Heaps
  - Gen 2 Collections
  - Gen0 Heap Size
  - Gen1 Heap Size
  - Gen2 Heap Size

- ASP.NET
  - Application Restarts
  - Applications Running
  - Request Execution Time
  - Request Wait Time
  - Requests Current
  - Requests Queued
  - Requests Rejected
- ASP.NET Applications
  - Anonymous Requests
  - Anonymous Requests/sec
  - Requests Executing
  - Requests Failed
  - Requests Timed Out
  - Requests Total
  - Requests/Sec
  - Sessions Active
  - Sessions Timed Out
  - Sessions Abandoned
  - Sessions Total

## Monitoring Databases and Java Application Servers

Databases support Performance Monitor counters on Windows. WebLogic, Tomcat, and WebSphere do not. They have their own methods of monitoring, see their documentation.

## Plumtree Analytics Server

Plumtree Analytics Server is an advanced usage tracking and analytics tool designed exclusively for the Plumtree portal. This portal add-on enables you to assess portal ROI and

define future opportunities with usage trends in mind. Plumtree's Analytics Server delivers the following features out-of-the-box:

- **Usage Tracking Metrics:** Analytics Server will plug directly into the Portal engine to retrieve metrics for common portal functions including community, portlet, and document hits as well as search queries, logins and more.
- **Behavior tracking:** Track usage patterns such as community visits. Also, find out the duration of portal use.
- **User Profile Correlation:** Correlate metric information with user profile information. In this way, usage tracking reports can be viewed and filtered by profile data such as country, company and department.

Plumtree Analytics Server includes the following reports that can be customized by setting filtering, grouping, and presentation options.

Table 5-2: Analytic Server Reports

Report	Description	Features
Community Traffic	Displays traffic information for each community in the portal.	Traffic is displayed in three ways: <ul style="list-style-type: none"><li>• Hits: Count of page views within the community.</li><li>• Visits: Count of visits to the community, each visit can consist of several hits.</li><li>• Users: Count of unique users who have visited the community. Users can select to see the most active, least active, or a select list of communities.</li></ul>



Table 5-2: Analytic Server Reports

Report	Description	Features
Community Response Time	Displays average, maximum and minimum response time for each community within the portal.	The response time is calculated as the time between the portal receiving a community page request until the time an HTML response is sent to the client. Users can select to see the slowest response times, fastest response times or response times for a select list of communities.
Portlet Usage	Shows usage statistics within gatewayed portlets.	<p>The traffic is displayed in two ways:</p> <ul style="list-style-type: none"><li>• Activity: Count of hits on an object (for example, a button or link) within a portlet.</li><li>• Users: Count of unique users who have performed an activity within the portlet.</li></ul> <p>Users can select to see the most active, least active, or a select list of portlets.</p>
Portal Traffic	Shows an aggregate of all portal page views within the portal.	

Table 5-2: Analytic Server Reports

Report	Description	Features
Portal Users	Displays statistics regarding portal user accounts.	<p>The following four figures are displayed to help explain user inception and activity.</p> <ul style="list-style-type: none"><li>• Total: Total user accounts in the portal.</li><li>• Added: Added (new) user accounts created in the portal during a given date range.</li><li>• Active: Active users defined by activity during a given date range.</li><li>• Inactive: Inactive users defined by inactivity during a given date range</li></ul>
Portal Logins	Shows an aggregate of all portal logins.	
Portal Duration	Displays the length of visits to the portal.	Visit durations are calculated as the time between login and logoff or the time between login and inactivity for a configurable length of time. This report shows both average and maximum visit duration.
Search Keywords	Shows the top search keywords entered in searches within the portal.	Select to see top 5, 10, 25, 50 or 100 search keyword phrases entered within the portal.

Table 5-2: Analytic Server Reports

Report	Description	Features
Document Views	Shows statistics for document views in the portal.	<p>These statistics can be displayed in two ways:</p> <ul style="list-style-type: none"><li>• Top Documents: List of top documents viewed with view count.</li><li>• Folders: Count of all document views by folders in the knowledge directory.</li></ul>

## PTSpy

PTSpy is a tool for debugging the Plumtree Corporate Portal. It extracts debugging and general state information from the portal as it runs. This information is critical in understanding problems and configuration errors.

PTSpy is highly tuned and can be run on a live Portal Server without adding significant overhead. It provides additional features including fine-grained filtering, opening of saved log files, highlighting of errors, find, and sort.

Adding more trace types decreases performance; however, leaving all warning and error traces on in a live environment does not impact the overall performance. Typically you can leave the portal running and logging these messages to a file. You can later look at them if needed.

A good strategy is to always leave warning and error traces enabled and log them to a file. That way, when a problem does occur, you will have a good sense about what the problem was. Often this is enough information to fix the problem.



**Note:** Simply running PTSpy is not intrusive, but advanced features of PTSpy are intrusive. Some advanced features of PTSpy will change the state of your portal. For example, under the Gateway component, you can **Disable Gateway Caching**. If you check this box, and then close PTSpy, gateway caching will remain disabled. When you use advanced features of PTSpy, be sure to reset PTSpy (in the Runtime

Settings dialog box) when you are finished debugging. For information on runtime settings, refer to [“Runtime Settings” on page 5-20](#).

## Using PTSpy

To use PTSpy, follow these steps:

1. Determine the component on which you should run PTSpy, such as the Automation Server or the Portal Server.

If the malfunction requires a Job to operate, such as a Crawler or a Publication, run PTSpy on the Automation Server. If the error occurs while using some part of the Web user interface, run PTSpy on the Portal Server.

2. To run PTSpy, click **Start | Programs | Plumtree | PTSpy**.

Running PTSpy with the default settings can usually give you some insight into a problem. If more detail is necessary, change the runtime settings by clicking **View | Runtime Settings**. For information on runtime settings, refer to [“Runtime Settings” on page 5-20](#). If you need help determining which settings to turn on or off, contact Plumtree Customer Support.

3. Reproduce the problem you experienced in isolation.
  - If the problem is with a Job, set the Job to run while no other Jobs are occurring.
  - If the problem is an error message, simply reproduce the error.

PTSpy logs the function calls made during the operation.

4. Review the PTSpy output (either in the PTSpy window or in the log file) and look for errors that might be related to your problem. For information on PTSpy output, refer to [“PTSpy Output” on page 5-21](#).

Search for the following words:

- error (generic indicator of errors)
- ADO error (indicates problem connecting to Plumtree database)
- Java stack trace (generic indicator of errors)
- timeout (indicates problem connecting to any one of the Plumtree Servers)

If you locate errors in the PTSpy output, review the context surrounding errors for other irregularities. PTSpy can help to determine what your errors are and during which function they occurred.

The following is an example of output from the PTSpy output:

```
ADO Error [0]: Number = 0x80004005, NativeError = 0x6, Source =
Microsoft OLE DB Provider for ODBC Drivers, Description =
[Microsoft][ODBC SQL Server Driver][Nam
[TRACE, PID 458, TID 546, 14:34:35] ed Pipes]Specified SQL server
not found., SQLState = 08001
```

Find a word that signals a problem, such as “error.” In this example, “ADO Error” signifies that there is a problem connecting to your Plumtree database. This information leads you to the next step in the troubleshooting process, which in this case would be checking the ODBC connection. Read the “ODBC Testing” section for more information.

For solutions to errors, search the Plumtree Knowledge Base or contact Plumtree Customer Support.

5. When you are done investigating problems, reset PTSpy to its defaults. Click **View | Runtime Settings**, then, in the Runtime Settings dialog box, click **Reset**. PTSpy manipulations can change the way the product works, in areas as low-level as gateway caching. Resetting PTSpy guarantees that the portal will operate normally after your investigation is finished.

## Runtime Settings

Runtime settings allow you to turn on or off each of the different trace types for each of the components. To access the runtime settings, click **View | Runtime Settings**.

By default the runtime settings are categorized by trace type. Each trace type can be turned on or off individually:

- Debug-severity traces the finest details and is often useful for developers to get detailed information about problems.
- Info-severity traces are used to give general expected information about the system. This is the standard trace.
- Warn-severity traces provide information on things out of the ordinary, but that are not necessarily a problem. Frequently warnings can be removed by changing configurations. For example, portlets that time out will trace warnings. Increasing the default timeout values for the portlets will make it more frequently appear for users.
- Error-severity traces are problems with the system that should be corrected.
- Fatal-severity traces are critical errors that prevent the portal from running. These are always enabled.
- Function tracing is similar to debug-severity tracing in that it is often useful for developers. It peeks into how the code is executed.
- Performance tracing gives measurements on how long certain activities take and can be used in tuning.
- Action tracing shows when a new event is being performed, such as a new incoming request to the portal beginning to be serviced or finishing being serviced. This tracing level is on by default (as are warn, error, and fatal). It can be used to find out what specific action caused a warning or error. For example, a warning might not include much information about what caused the warning, but if you see the previous action trace on that thread, it shows the URL accessed which caused the problem.

Changing the runtime settings view to **Component** shows each of the different parts of the portal. Each component can be turned on or off individually. This view contains a few more options than the trace type view, but the two of them can generally be considered a matrix for turning on or off each of the different trace types for each of the components.

## PTSpy Output

PTSpy has several features to help you find information of interest within its output:

- Find (Ctrl+F)
- Color highlighting of various trace levels (for example, blue for warning, red for error, and a background highlighting for fatal errors)
- Column sorting

PTSpy captures all trace messages across all processes and threads. Typically these overlap each other with several threads all tracing at the same time. To highlight all the rows for a particular thread, making them stand out, click **Edit | Highlight from Same Thread**.

Sometimes errors frequently appear and have filled a log. If you have several thousand traces, sorting by Type gives a good estimate about the frequency of a problem. Several of the same warning or error traces in sequence usually indicates a problem.

## View Source

HTML code creates Web pages. In turn, Plumtree Activity Spaces generate HTML code. Along with HTML from the View and Display pages, the underlying framework inserts some general information for each page. If there is an error on the page, the Error framework might insert additional debugging information. You can review the HTML source for any given Web page to gather this information. Often the HTML for a given error page contains detailed information about the error.

### When to Use View Source

Use View Source to gather more information when you receive an error on a portal page or when you want some general information about the page. For example, use View Source if you receive the following error message on a portal page: "An unexpected error occurred when trying to start the Editor." The message itself gives no clues to the source of the error, but when you view the HTML source code for the page, you might be able to determine the source of the error.

### How to Use View Source

While viewing the Web page, in the browser menu, click **View | Source**. This displays the HTML for that page. If the browser menu is unavailable, sometimes it is possible to view source by right-clicking the page, and then clicking **View Source**. With this approach, be aware that if there are frames, only the source for the frame in which you right-clicked will display. When the source displays, you can search for specific pieces of information as described in the next section.



## What Is Available in View Source

Each portal page contains several pieces of general information:

- To determine the server hosting the portal, search for "Hostname:". The hostname of the server is commented in the source: "`<!--Hostname: MyServer-->`".
- To find information about the build of the portal, search for "Portal Version:", "Change-list:", and "Build Date:".
- To find information about general timing data points, search for "Total Request Time:", "Control Time:", "Page Construction Time:", and "Page Display Time:".

If there is an error on the page, View Source might provide extended information. There are three items that you can search for

- To view the error, search for "alertErrorTitle". You might have to repeat the search because several error related HTMLElements might use that text.
- To view extended information, search for "Extended Error Message:". The extended error is wrapped in an HTMLComment and thus does not show up on the page, "`<!--Extended Error Message: Sample Extended Error message.-->`". The extended information, controlled by the developer and Activity Space, is frequently the same as the error message that displays in the user interface.
- You might also need to search for "unexpected error". When the portal encounters an unexpected error, the stacktrace for the error is often inserted into an HTMLComment. The following example informs the user where the error originates from. The user then has a starting point from which to perform further debugging:

```
<!--An unexpected error occurred when trying to start the
Editor.:
com.plumtree.openfoundation.util.XPEException: An unexpected error
occurred when trying to start the Editor.
at com.plumtree.portalpages.admin.editors.group.Group-
Model.DoTaskOnStartEditor(GroupModel.java:411)
```

## ODBC Testing

### When to Use ODBC Testing

Test your ODBC connection when you experience logon errors, such as the error, “Cannot log on as a guest user.” If you do not have a successful connection to the Plumtree database, you will not be able to log on to your portal. The Plumtree Data Source Name points to the Plumtree database and creates a database connection. When you cannot log on, you should test the Data Source Name to make sure it connects to the database.

### ODBC Testing: SQL Server

Use ODBC testing to evaluate the DSN (Data Source Name) that you created for your Plumtree database. It must be configured correctly (you must also confirm that the DSN is pointing to the Plumtree database). You can run a simple test to see if the DSN is configured properly. If this test shows that the DSN is not properly configured, you must reconfigure the DSN. The following procedure includes steps to reconfigure a DSN.

#### *How to Test a SQL Server Database Connection*

If you are running SQL Server, test the ODBC Data Source Name by following these steps:

1. Click **Start | Programs | Administrative Tools | Data Sources (ODBC)**.
2. On the System DSN tab, click the Plumtree Data Source, and then click **Configure**.
3. Make sure the **Server** drop-down list shows the machine that houses your Plumtree database, and then click **Next**.
4. Make sure **With SQL Server authentication using a login ID and password entered by the user** is selected. Make sure the **Login ID** and **Password** are correct for the Plumtree database, and then click **Next**.
5. Make sure **Change the default database to** is selected and that the Plumtree database is selected in the drop-down list, and then click **Next**.
6. Click **Finish**.

7. In the ODBC Microsoft SQL Server Setup dialog box, click the **Test Data Source** button to make sure your Plumtree DSN is properly configured.

You will receive a “Tests Completed Successfully” message if you have a connection to your Plumtree database using the Plumtree DSN.

If you do not receive a “Tests Completed Successfully” message, your Plumtree DSN is not pointing to your Plumtree database. Configure the DSN correctly by repeating steps 2 through 6 to double-check that you have selected the machine that houses your Plumtree database in step 3. Also, make sure you selected the **SQL Server authentication** option and provided the correct Login ID and password in step 4. The most common reason for a lack of connection to the Plumtree database is an incorrect Login ID and password for the Plumtree DSN.

## ODBC Testing: Oracle

Testing Oracle is conceptually very similar to testing SQL Server. You choose your DSN, and use the Test facility. These instructions are based on the assumption that you are using the Plumtree Data Direct Oracle Driver (in the ODBC Data Source Administrator, the Driver column of the relevant row on the System DSN tab should say "Plumtree Data Direct Oracle Driver.")

### *How to Test an Oracle Database Connection*

If you are running Oracle, test ODBC Data Source Name by following these steps:

1. Click **Start | Programs | Administrative Tools | Data Sources (ODBC)**.
2. On the System DSN tab, click the Plumtree Data Source, and then click **Configure**.
3. On the General tab:
  - a. Make sure the **Host** is set to the name or IP address of the machine on which Oracle is running.
  - b. Make sure the **Port Number** is set to the correct port number for your Oracle installation.
  - c. Make sure the **SID (System Identifier)** is set to the correct SID for your Oracle installation.

Contact your Oracle Administrator to obtain the correct values.

4. Click the **Advanced** tab.
  - a. Make sure the **Default User Name** is set to your Plumtree database user name.
  - b. Make sure the **Local Timezone Offset** box is blank.
  - c. Make sure the **Default Buffer Size for Long/LOB Columns** is set to 1024.
  - d. Make sure the only check boxes that are checked are **Application Using Threads** and **Enable N-CHAR Support**. These options must be selected.
  - e. Make sure "0" is selected in all three drop-down lists.

5. Click the **Performance** tab.
  - a. Make sure the **Array Size** is set to 60000.
  - b. Make sure the **Lock Timeout** is set to -1.
  - c. Make sure **User Current Schema for SQLProcedures**, **Catalog Functions Include Synonyms**, and **Enable Scrollable Cursors** are checked.
  - d. Make sure **Enable Static Cursors for Long Data** is not checked.
  - e. Make sure the **Cached Cursor Limit** is set to 32.
  - f. Make sure the **Cached Description Limit** is set to 0.
6. Click the **Failover** tab. It should be blank; failover is not supported.
7. Click the **About** tab. Make sure you are using the correct version of the drivers (refer to the Portal installation guide).
8. Click **Test Connect**.
9. In the dialog box, type the user name and password that the Portal will use to connect to Oracle and click **OK**. You should receive the message "Connection Established!" If the test is successful, you have a connection to your Oracle Plumbtree database. If the test fails, you do not have a connection. Consult your Oracle DBA.

## Testing and Troubleshooting Enterprise Web Content

This section will help you troubleshoot common issues with portal content. Portal content refers to all content in the Knowledge Directory.

### General

- Search Update Agent (SUA)
  - The SUA is an intrinsic portal operation that indexes all objects in the portal including documents in the Knowledge Directory. Changes made to objects in the portal will not be reflected in search results until an SUA finishes. The SUA is run by a series of scheduled jobs. By default, these jobs are set to run every 10 minutes, but you can alter this frequency. Ensure that the time between SUA jobs is longer than the actual time that it takes to complete a single run of the search update agent. For example, if the search update agent takes 5 minutes to run, make sure that the time between SUA jobs is longer than 5 minutes. If the time between SUA jobs is shorter than the time to complete an SUA job, the job will never reach the completed state.
  - You can run as many SUAs as you want, but they must be in separate jobs. Separate SUA jobs can run concurrently. For example, if a single SUA takes 5 minutes to complete, You can start one SUA to run at 9:00 AM and create a second SUA job that starts at 9:02 AM. The time between *individual* jobs should still be greater than 5 minutes, for example, run every 10 minutes.
  - When a user performs a portal search via the portal banner or through advanced search, the portal returns results based on the Search Server index created by the SUA.
- When you browse the Knowledge Directory, you get an error stating that “the Search Server could not be contacted.”

The portal allows Knowledge Directory folders to be browsed in two ways. One is via search results and the other is with database calls. Browsing via search looks exactly the same as browsing via the database. The benefit of browsing via search is that it reduces the load on the database. The benefit of browsing via the database is that the

data is always the latest data available (remember that changes will not be reflected in search results until the SUA runs).

- Ensure that the Plumtree Search Server is running.
- Make sure that the portal is connected to the correct Search Server and that the connection information is correct. Connection information is specified in the Search Server Manager utility, accessed through the Administration page.
- To switch to browsing via the database, go to the Knowledge Directory Preferences utility, accessed through the Administration page, and switch “Browsing Source:” to Database.
- How do you gateway a document in the Knowledge Directory?

When a document is gatewayed the user's Web browser retrieves the document via the portal. This prevents the user's Web browser from having to connect to the computer where the document is stored. This is beneficial in the case when external users need to access documents on internal computers.

- Ensure that the data source is configured to gateway content. In the administrative object directory, open the data source. On the Main Settings page, select “Uses the Gateway to open documents” in the URL Type section.
- If you are gatewaying Content Server content, ensure that the Content Server path specified in the crawler is identical to the gateway URL prefix path specified on the Content Server Crawler Web Service - HTTP Configuration page. For example, if the gateway URL prefix path in the web service is set to `http://contentserver.mycompany.com/ptcs/publishedcontent/public`, the path in the crawler must begin with this path. In this case, setting up the crawler to crawl `http://contentserver/ptcs/publishedcontent/public/employee_profiles/`, will result in non-gatewayed content.
- Once a document is crawled into the portal as non-gatewayed it cannot be modified to be gatewayed and vice versa. The document must be deleted and re-crawled in appropriately.

## Crawlers

- My crawler is not bringing any documents into the portal.
  - Verify that the job runs. Open the job history and ensure the last run date is the expected date. If the job did not run, check that the folder in which the job is stored is registered in the Automation Server utility.
  - Verify that the crawler job runs without errors. The status for the last run job should be “Succeeded”.
  - If the job status is “Failed”, search for the word “error” in the job history log and solve the error appropriately.
  - Does the crawler destination folder use filters? If so, make sure the filter is not impeding the documents crawled in. If this is the case, the job history log will inform you that the documents were not crawled in due to a filter restriction.
  - Check the data source and ensure that the document types that you aim to crawl are included in the data source definition.
  - Certain crawlers can exclude or include specific files. For Web crawlers, the crawler can be set to only include URLs of a specific format or to exclude URLs of a specific format. Check to make sure that the inclusion/exclusion properties are correctly set.
  - Ensure that the user the data source impersonates has access to the source files that you want to crawl. When you select a source folder to crawl from, the portal uses the editing user's network identity to determine which folders you can choose from. When performing a crawl, the crawler uses the data source user to determine what files it can access. The editing user and the data source user might have different rights to the source files.
  - If the crawler has run previously, it could be that a particular document has already been deleted or rejected. By default, crawlers are set to not bring in already crawled documents. If you would like to re-crawl a previously deleted or rejected document, the crawler can be set to do so on the Advanced Settings page of the crawler. However, the crawler will re-crawl all previously deleted or rejected documents.



- The documents crawled in by the Plumtree Crawler Web Service for NT (NT CWS) do not have properties associated with them.
  - For Microsoft Office documents the NT CWS uses Office applications API to retrieve document properties. Therefore, ensure that Microsoft Office is installed on the machine hosting the NT CWS and that the Office applications (for example, Word, Excel, PowerPoint) can be run.

## Portal Search

- A folder or document is returned in search results but you get an error when you try to access it.

Changes to the ACL of an object are updated in the search collection by the SUA. If the SUA has not run, there might be a lag where an object that the user does not have access to appears in a search result.

- Make sure the SUA has run.
- I secured a folder to restrict access to a branch of the Knowledge Directory hierarchy, but subfolders and documents still appear in search results.

Users only need Read access to folders for them to appear in search results.

- Secure each folder in the Knowledge Directory hierarchy.

## Portal Search/Snapshot Queries Portlet

- A user searches for a document and cannot find it or a user does not see a document in a snapshot query portlet or “No results were returned for this search” error displayed in snapshot query portlet.
  - Is the document approved? In the Knowledge Directory, while in Edit mode, you will see a red dot next to documents that are not approved. If the document is not approved, check the box next to the document and click **Approve** in the menu above.

- Has the SUA run since the document was crawled in or submitted? An easy way to find out if a document has been indexed by the SUA is by browsing the Knowledge Directory. If you are able to find the document by browsing the directory (not in Edit mode) then the document has been indexed.
- If you have modified the security of the document recently, has the SUA run since the modification was made? Changes to document security do not take effect immediately. Instead, the SUA re-indexes documents whose security has been modified.
- Verify that the user has Read access on both the document and the parent folder.
- A snapshot query's cache is refreshed every 15 minutes. Therefore, after changing access to documents referenced by a snapshot query, the snapshot query results can still display these documents for up to 15 minutes.
- A user sees different information in a snapshot query than what you see.

Snapshot queries use search technology to generate the query results. When previewing a snapshot query during the creation process, the preview displays the results that the creator can see based on the snapshot query properties. This is done for security purposes. However, when the snapshot query is displayed from a My Page or community, the snapshot query displays what the end-user can see based on the snapshot query properties. Because of security, this might be different than what the creator can see.

- “Page cannot be found” error when clicking on a document in a search results page.
  - This error generally happens when the user's machine does not have access to a non-gatewayed document. For example, an internal host name is used to crawl a document into the portal through a data source that is not set up to gateway content. If this is the case, see the previous bullet on “How do you gateway a document in the Knowledge Directory?”.
  - If the document is accessed by search (for example, search results or browsing via search), the document might be deleted but the search index has not been updated. Run the SUA to update the search index.

- “Gateway was not able to access requested content. If the error persists, contact your portal administrator.” error when clicking on a document in a search results page.
  - Verify that the file exists in the file store from which it was crawled. To find the original file path, view the document properties. You will need to manually decode the File URL property as it is HTTP encoded.
  - Verify that the File URL property and the actual path in the file store are identical. If any of the folders leading to the file have been renamed (or the file has been renamed), the portal will not be able to locate the file and will produce the gateway error message.
  - Verify that the file can be opened from the Portal Server. From the Portal Server, open Windows Explorer and browse to the network path where the file is stored. Open the file locally. If the file does not open, in most cases a network problem or security problem is preventing the portal from accessing the file.
  - Check PTSpy for useful error information. With PTSpy running, reproduce the gateway error on the portal. Then, review the messages produced by PTSpy.
  - Users need at least Read access to all of the components that make up a document. A document uses a data source. A data source uses a web service. A web service might use a remote server. Users need at least Read access to all of the dependent objects in order to view the document.

## Search Server Maintenance

This section provides suggestions for maintaining your Search Server.

### Logs

The Search Server provides logs to help you monitor Search Server performance and to diagnose problems.

### System Events

The Search Server performs self-diagnostics at startup and will log any errors or warnings to the system event logs. Refer to Plumtree Knowledge Base article DA\_207257: “5.0 Search Server Event Log Errors and Warnings” for more information.

### Search Server Logs

The Plumtree Search Server generates log files on disk. The logs contain information regarding queries processed, documents indexed, and Search Server performance. This information is processed by the Search Log Analysis external operation within the 5.0 portal and can also be inspected manually to monitor Search Server usage or to diagnose problems with search. Refer to Plumtree Knowledge Base article I1473: “Plumtree Search Server Log Files” for more information about Search Server log files. Refer to Plumtree Knowledge Base article DA\_131654: “5.0 Search Server Log Reports” for more information on search log reports.

### Log Rotation

Versions of the Search Server shipping with Plumtree Corporate Portal, version 5.0.2 and later include the ability to perform automatic log rotation. Old log files are automatically deleted when new ones are created. This feature is disabled by default, causing log files to be retained indefinitely. Refer to Plumtree Knowledge Base article DA\_207213: “Search Server Log File Rotation” for information on how to enable and configure this feature.

## Status Monitoring

The status of the Search Server can be checked manually from the Search Server Manager utility (accessed through the Administration page). Click **Show Status** to send a status request to the Search Server and display some basic server statistics, such as uptime and number of searchable items in the index. Clicking **Show Status** also causes the Search Server to log status and performance information (discussed previously).

Because the Search Server is a TCP service listening on a port, any standard network monitoring system can monitor its status automatically. The recommended method is to open a TCP connection to the search service and send the string "PING\n" as a request. A healthy Search Server will respond with an XML message like:

```
<request client="10.1.0.65" duration="0" latency="0" livequeries="1"
lockLatency="0" thread="236" type="ping">
  ALIVE
</request>
```

The client IP and other request attributes will vary. If the Search Server is unavailable, the connection will fail. For optimal networking performance, the monitoring tool should consume the response text (read until EOF) from the socket before closing it.

## Performance Monitoring

The I/O and computation required to process search indexing and query requests is split between the Search Server and the client (Portal Server, Automation Server, or portal application). For most requests, the bulk of this processing occurs on the Search Server, while the client adds some overhead for text processing, request marshalling, UI rendering, and so on. Refer to Plumtree Knowledge Base article 11470: "Document Text Extraction for Search" for more information on the text extraction process. The Search Server logs contain timing information for each query or indexing request. In addition search status requests cause performance summary information to be logged to the Search Server logs. The amount of timing and performance information varies depending on the verbosity setting for the Search Server logs. Generally Search Server administrators will only be interested in the overall duration for each request, which is logged at any verbosity level. Refer to Plumtree Knowledge Base article 11473: "Plumtree Search Server Log Files" for more information about the contents of Search Server log files.

## Backup and Restore

The Search Server deployment includes a Replicate utility that can be used for online backup and restore of the search collection. Third-party backup software can be used in conjunction with the Replicate utility, but should not be used directly against the Search Server archive collection files. Refer to Plumtree Knowledge Base article 11875: “Replicating a 5.0 Search Collection” for more information on Search Server backup and restore.

## Repair

### Index Synchronization with Portal

The Search Update Agent (SUA) is responsible for keeping the search index synchronized with the portal database. The SUA manages incremental changes to the portal database and indexing and updating the search index with recent changes to objects in the Knowledge Directory and Administrative Object Directory. Normally, this incremental synchronization is sufficient to keep the search index and portal database closely synchronized. Occasionally, however, errors may occur during indexing jobs that cause the contents of the search index to diverge from the contents of the portal database. The Search Repair process is intended to correct this kind of divergence and is normally scheduled to run once a week. Refer to the *Administrator Guide for the Plumtree Corporate Portal* for more information about the Search Update Agent and Search Repair.

### Index Corruption - Self Repair

The Search Server performs self-diagnostics at startup and will log any errors or warnings to the system event logs. In cases where the Search Server can automatically correct the problem, a warning is logged and the self-repair process is initiated. In cases where manual intervention is required, an error is logged, and the Search Server startup will fail. Refer to Plumtree Knowledge Base article DA\_207257: “5.0 Search Server Event Log Errors and Warnings” for more information.

## Index Corruption - Manual repair

Examinearchive is a Plumtree Search Server utility that can be used to examine, test, and repair the Search Server index while the server is stopped. You do not need to use examinearchive for routine maintenance. Typically you will only use examinearchive if instructed to do so by Plumtree Customer Support. Refer to Plumtree Knowledge Base article 11596: “Examinearchive User's Guide” for more information.

## Rebuild Search Index

Plumtree Knowledge Base article DA\_199161: “HOWTO - Rebuild Search Server Index” describes how to clean and rebuild the search index from scratch.

