Sun Java System Portal Server 7.1 Deployment Planning Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–5073
01/25/2007

Copyright 2007 Sun Microsystems, Inc.   4150 Network Circle, Santa Clara, CA 95054 U.S.A.   All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2007 Sun Microsystems, Inc.   4150 Network Circle, Santa Clara, CA 95054 U.S.A.   Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE “EN L'ETAT” ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

070312@17039

Contents

Preface 17

1 **Introduction to Portal Server Deployment Planning** 21
About Portal Server 21
Portal Server Features 22
Identity Management 22
Secure Remote Access 24
Search Service 26
Content Personalizing 26
Aggregation and Integration 27
Using The Solution Life Cycle 28
Solution Life Cycle 29

2 **Understanding Your Business Goals** 31
About Business Analysis 31
Defining Business Requirements 32
Understanding User Behaviors and Patterns 33
Facilitating Productivity 34

3 **Identifying Your Technical Requirements** 35
About Technical Requirements 35
Usage Analysis for Portal Server 36
Peak Numbers 36
Average Time Between Page Requests 37
Concurrent Users 37
Average Session Time 38
Search Service Factors 39

Page Configuration 39
Portal Desktop Configuration 40
Hardware and Applications 40
Back-End Servers 41
Transaction Time 41
Workload Conditions 41
Usage Analysis for Secure Remote Access 42
Scalability 43
Using Secure Remote Access Prototype Numbers 43
Key Performance Factors 43
Portal Server Use Cases 45
Preparing to Develop Portal Server Use Cases 45
Example Use Case: Authenticate Portal User 46
Quality of Service Requirements 47
Performance 48
Availability 48
Scalability 48
Security 49
Designing for Localization 51

4 Designing Your Logical Architecture 53
Analysis of Logical Architecture 53
High-Level Logical Architecture 54
Low-Level Logical Architecture 54
Portal Server Components 55
Secure Remote Access Components 57
Secure Remote Access Gateway 57
Netlet 60
Netlet Proxy 62
NetFile 62
Rewriter 65
▼ To Route Secure Remote Access Requests 65
Rewriter Proxy 66
Proxylet 66
Portal Server Nodes 67

Portal Server and Access Manager on Different Nodes 67
Portal Server System Communication Links 68
Example Portal Server Logical Architectures 71
A Typical Portal Server Installation 71
Portal Server Building Modules 73
Access Manager and Portal Server on Separate Nodes 80
Two Portal Servers One Access Manager 81
Two Portal Servers Two Access Managers 81
One Load Balancer Two Access Managers 82
Example Secure Remote Access Logical Architectures 83
Basic Secure Remote Access Configuration 83
Disable Netlet 84
Proxylet 85
Multiple Gateway Instances 86
Netlet and Rewriter Proxies 87
Netlet and Rewriter Proxies on Separate Nodes 89
Two Gateways and Netlet Proxy 89
Gateway with Accelerator 90
Netlet with Third-Party Proxy 91
Reverse Proxy 92
Deployment Scenario 92

5 Planning Your Deployment Design 93
About Deployment Design 93
Estimating Processor Requirements 94
▼ Steps to Estimate Processor Requirements 95
Customize the Baseline Sizing Figures 95
Validate Baseline Sizing Figures 96
Refine Baseline Sizing Figures 96
Validate Your Final Figures 97
Identifying Performance Bottlenecks 97
Memory Consumption and Garbage Collection 97
Optimizing Resources 98
SSL Off-loading onto Load Balancer 98
Sun Enterprise Midframe Line 98

Managing Risks 99

6 **Implementing Your Deployment Design** 101
About Implementing Deployment Designs 101
Documenting the Portal 102
Developing a Portal Prototype 102
▼ To Develop a Portal Prototype 102
Access Manager Cache and Sessions 104
Thread Usage 104
Portal Usage Information 105
Identity and Directory Structure Design 105
Creating a Custom Access Manager Service 106
Portal Desktop Design 107
Choosing and Implementing the Correct Aggregration Strategy 107
Working with Providers 108
Software Deployment 109
Software Packaging 109
Software Categories 109
Compatibility With Java Software 110
Client Support 110
Content and Design Implementation 111
Placement of Static Portal Content 111
Integrating Applications 112
Rolling Out a Production Deployment 113

7 **Deploying Portal Server Communities** 115
Deploying Communities on the Default Database Installation 115
Deploying Communities on Multiple Hosts Using a Single Database Instance 117
Deployment Requirements 118
▼ To Set Up a Common Derby Network Server Instance (Solaris) 118
▼ To Point All Portal Server Instances To The Common Derby Network Server Instance . 120
Deploying Communities on a High-Availability Database 124
▼ To Replace the Derby Database 124
▼ To Deploy High-Availability Database 126

8 Basics of Portal Performance 131
Performance Constraints 131
Performance Drivers 132
Arrival Rates 132
User Activity 132
Desktop Channel Types 133
Portal Server Tuning 133
Web Container Tuning 133
Tuning Scripts 134
Access Manager Tuning 143
Directory Server Tuning 143
Monitoring CPU Utilization 143
Input 143
Output 144
What to Look For 144
Monitoring Memory Utilization 144
Input 144
Output 144

A Understanding Portal Server and Application Servers 145
Application Server Support in Portal Server 145
Portal Server on an Application Server Cluster 146
▼ To Install Portal Server on an Application Server 147
Application Server Enterprise Edition 147
BEA WebLogic Server 147
IBM WebSphere Application Server 149

B Troubleshooting Your Deployment 151
Troubleshooting Portal Server 151
Troubleshooting the Portal Server Installation 151
Checking Unix Processes 153
Runtime Log Files 154
Restarting Processes 155
▼ To Restart the Web Server 155
▼ To Restart Cacao 155

▼ To Restart the Directory Server ... 155
Working with the Display Profile ... 155
▼ To Extract the Display Profile ... 156
▼ To Reload the Display Profile ... 156
Configuring an HTTP Proxy ... 156
▼ To Configure Usage of an HTTP Proxy for a Portal Server Instance ... 156
Troubleshooting Secure Remote Access ... 157
▼ To Check Secure Remote Access Status ... 157
▼ To Enable Secure Remote Access After Installation ... 157
▼ To List the Secure Remote Access Instance ... 157
▼ To Start Secure Remote Access Instance ... 158
▼ To Provision Secure Remote Access Instance ... 158
▼ To setup Non-authenticated URL List for Secure Remote Access ... 158
Debugging the Gateway ... 159
▼ To Check the Gateway Process ... 159
▼ To Use Debugging ... 159
Secure Remote Access Log Files ... 160

C Portal Deployment Worksheets ... 161
Portal Assessment Worksheets ... 161
Portal Design Task List ... 164

D Understanding How Files and Directories Are Installed ... 171
Directories Installed for Portal Server ... 171
Directories Installed for Secure Remote Access ... 172
Log File Locations ... 173
Portal Server File Locations ... 173
Community Sample Files ... 174
Community Template Locations ... 174
Dynamic Content File Locations ... 174
Community Sample Portal Properties File Location ... 175
Static File Locations ... 175
Comparison of Solaris and Linux Path Names ... 176

Index 177

Figures

FIGURE 1–1 Solution Life Cycle 30
FIGURE 4–1 Portal Server Communication Links 69
FIGURE 4–2 High-level Architecture for a Business-to-Employee Portal 72
FIGURE 4–3 Secure Remote Access Deployment 72
FIGURE 4–4 Portal Server Building Module Architecture 73
FIGURE 4–5 Best Effort Scenario 75
FIGURE 4–6 No Single Point of Failure Example 76
FIGURE 4–7 Transparent Failover Example Scenario 78
FIGURE 4–8 Access Manager and Portal Server on Different Nodes 80
FIGURE 4–9 Two Portal Servers and One Access Manager 81
FIGURE 4–10 Two Portal Servers and Two Access Managers 82
FIGURE 4–11 Load Balancing two Access Managers 82
FIGURE 4–12 Basic Secure Remote Access Configuration 84
FIGURE 4–13 Disable Netlet 85
FIGURE 4–14 Proxylet 86
FIGURE 4–15 Multiple Gateway Instances 87
FIGURE 4–16 Netlet and Rewriter Proxies 88
FIGURE 4–17 Proxies on Separate Nodes 89
FIGURE 4–18 Two Gateways and Netlet Proxy 90
FIGURE 4–19 Secure Remote Access Gateway with External Accelerator 91
FIGURE 4–20 Netlet and Third-Party Proxy 91
FIGURE 4–21 Using a Reverse Proxy in Front of the Gateway 92

Tables

TABLE 1–1 Identity Management Features and Benefits 23
TABLE 1–2 Secure Remote Access Features and Benefits 24
TABLE 1–3 Search Features and Benefits 26
TABLE 1–4 Personalization Features and Benefits 27
TABLE 1–5 Aggregation Features and Benefits 28
TABLE 3–1 Use Case: Authenticate Portal User 46
TABLE 4–1 Portal Server High Availability Scenarios 74
TABLE C–1 General Questions 162
TABLE C–2 Organizational Questions 162
TABLE C–3 Business Service-level Expectations Questions 163
TABLE C–4 Content Management Questions 163
TABLE C–5 User Management and Security Questions 163
TABLE C–6 Business Intelligence Questions 163
TABLE C–7 Architecture Questions 164
TABLE C–8 Design Task List 165
TABLE D–1 Portal Server Directories 171
TABLE D–2 Portal Server Secure Remote Access Directories 172
TABLE D–3 Log File Locations 173
TABLE D–4 Portal Server File Locations 173
TABLE D–5 Comparison of Solaris and Linux Path Names 176

Examples

EXAMPLE B–1 Java Enterprise System Install Logs 152

Preface

The *Sun Java System Portal Server 7.1 Deployment Planning Guide* provides an introduction to planning and designing deployment solutions for Sun Java™ System Portal Server and Sun Java System Portal Server Secure Remote Access.

This guide assumes you have read the *Sun Java Enterprise System 5 Deployment Planning Guide* which presents basic concepts and principles of deployment planning and design, and discusses the solution life cycle, which encapsulates the phases and tasks of a deployment design project.

Who Should Use This Book

This guide is primarily intended for deployment architects and business planners responsible for the analysis and design of portal server deployments. This guide is also useful for system integrators and others responsible for the design and implementation of various aspects of an enterprise application.

Before You Read This Book

This guide assumes that you have read the *Sun Java Enterprise System 5 Deployment Planning Guide*.

How This Book Is Organized

This guide is based on a solution life cycle which describes the various phases of deployment planning. The *Sun Java Enterprise System 5 Deployment Planning Guide*.

Portal Server Documentation Set

The Sun Java System Portal Server documentation is available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML) formats. Both formats are readable by technologies for users with disabilities. The Portal Server documentation can be accessed here:

http://docs.sun.com/coll/1483.1

The following table lists the books included in the Portal Server 7.1 core documentation set.

Book Title	Description
Sun Java System Portal Server 7.1 Configuration Guide	Describes how to administer Portal Server 7.1 using the Portal Server administration console.
Sun Java System Portal Server 7.1 Installation Guide	Describes how to install Portal Server after Java Enterprise System has been installed.
Sun Java System Portal Server 7.1 Release Notes	Available after the product is released. Contains last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.
Sun Java System Portal Server 7.1 Technical Overview	Provides an introduction to Portal Server concepts, architecture, and components.
Sun Java System Portal Server 7.1 Developer Sample Guide	Provides information about how to create and deploy a Portal Server 7.1 developer sample desktop.
Sun Java System Portal Server 7.1 Developer's Guide	Provides a high-level overview of the Sun Java System Portal Server software APIs.

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Meaning	Example

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–2 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Accessing Sun Resources Online

The docs.sun.com web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819–5073.

CHAPTER 1

Introduction to Portal Server Deployment Planning

This chapter provides a brief overview of Sun Java™ System Portal Server and Sun Java System Portal Server Secure Remote Access. This chapter introduces the solution life cycle, which outlines the various steps for planning and designing enterprise software systems. See the *Sun Java Enterprise System 7.1 Deployment Planning Guide* for information on basic concepts and principles of deployment planning and design.

This chapter contains the following sections:

- "About Portal Server" on page 21
- "Portal Server Features" on page 22
- "Using The Solution Life Cycle" on page 28
- "Solution Life Cycle" on page 29

About Portal Server

Portal Server is a component of the Sun Java Enterprise System technology. Sun Java Enterprise System technology supports a wide range of enterprise computing needs, such as creating a secure intranet portal to provide the employees of an enterprise with secure access to email and in-house business applications.

The Portal Server product is an identity-enabled portal server solution. It provides all the user, policy, and identity management to enforce security, web application single sign-on (SSO), and access capabilities to end user communities. In addition, Portal Server combines portal services, such as personalization, aggregation, security, integration, and search. Unique capabilities that enable secure remote access to internal resources and applications round out a complete portal platform for deploying business-to-employee, business-to-business, and business-to-consumer portals. Sun Java System Portal Server Secure Remote Access provides secure remote access capabilities to access web- and non-web enabled resources.

Each enterprise assesses its own needs and plans its own deployment of Java Enterprise System technology. The optimal deployment for each enterprise depends on the type of applications that Java Enterprise System technology supports, the number of users, the kind of hardware that is available, and other considerations of this type.

Portal Server is able to work with previously installed software components. In this case, Portal Server uses the installed software when the software is an appropriate version.

Portal Server runs in open mode or secure mode. In secure mode, it uses Secure Remote Access. The main difference between an open portal and a secure portal is that the services presented by the open portal typically reside within the demilitarized zone (DMZ) and not within the secured intranet. Secure Remote Access offers browser-based secure access to portal content and services from any remote browser enabled with Java technology. Integration with Portal Server software ensures that users receive secure encrypted access to the content and services that users have permission to access.

Secure Remote Access is targeted toward enterprises deploying highly secure remote access portals. These portals emphasize security, protection, and privacy of intranet resources. The Secure Remote Access services–Access List, the Gateway, NetFile, Netlet, and Proxylet– enable users to securely access intranet resources through the Internet without exposing these resources to the Internet.

Portal Server Features

This section reviews specific technology features with the goal of determining which technologies are most important for your organization. Review these features while keeping in mind your organization's short, mid, and long term plans.

Use the following sections and tables to assess the benefits of the listed features and determine their relative priority for your organization. This information will assist you in developing a deployment plan in a timely and cost effective manner.

Note – If your Java Enterprise System sales representative has previously discussed these topics with you, this section serves as a review of that process.

Identity Management

Portal Server uses identity management to control many users spanning a variety of different roles across the organization and sometimes outside the organization while accessing content, applications and services. The challenges include: Who is using an application? In what capacity do users serve the organization or company? What do users need to do, and what should users be able to access? How can others help with the administrative work?

Table 1–1 shows the identity management features and their benefits.

TABLE 1–1 Identity Management Features and Benefits

Feature	Description	Benefit
Directory service	Portal Server uses Access Manager and Directory Server	Portal Server uses an LDAP directory for storing user profiles, roles, and identity information for authentication, single sign-on (SSO), delegated administration, and personalization.
User, policy, and provisioning management	Access Manager enables you to manage many users spanning a variety of different roles across the organization and sometimes outside the organization while accessing content, applications, and services.	Provides central storage and management of identity information, which is integrated with a policy solution to enforce access rights. Extends a common identity to handle new applications, enables applications to share administrative work, and simplifies tasks normally associated with building these services. Consolidates management of users and applications. Personalizes content and service delivery. Simplifies and streamlines information and service access. Reduces costs associated with managing access and delivery. Provides secure policy-based access to applications. Ensures secure access as portal deployments expand beyond employee LAN access.
Single sign-on (SSO)	Access Manager integrates user authentication and single sign-on through an SSO API. Once the user is authenticated, the SSO API takes over. Each time the authenticated user tries to access a protected page, the SSO API determines if the user has the permissions required based on their authentication credentials. If the user is valid, access to the page is given without additional authentication. If not, the user is prompted to authenticate again.	Enhances user productivity by providing a consistent, centralized mechanism to manage authentication and single sign-on, while enabling employees, partners and customers access to content, applications, and services.

TABLE 1–1 Identity Management Features and Benefits *(Continued)*

Feature	Description	Benefit
Delegated administration	The Portal Server administration console provides role-based delegated administration capabilities to different kinds of administrators to manage organizations, users, policy, roles, channels, and Portal Desktop providers based on the given permissions.	Enables IT to delegate portal administrative duties to free up valuable IT resources and administration.
Security	Provides single sign-on for aggregated applications to the portal.	Security can address many different needs within the portal, including authentication into the portal, encryption of the communications between the portal and the end user, and authorization of the content and applications to only users that are allowed access.

Secure Remote Access

Table 1–2 shows the Sun Java System Portal Server Secure Remote Access features and their benefits.

TABLE 1–2 Secure Remote Access Features and Benefits

Feature	Description	Benefit
Integrated security	Extranet or Virtual Private Network capabilities "on demand" while providing user, policy, and authentication services. The Gateway component provides the interface and security barrier between remote user sessions originating from the Internet, and your corporate intranet.	Extends an enterprise's content, applications, files, and services located behind firewalls to authorized suppliers, business partners, and employees. To prevent denial of service attacks, you can use both internal and external DMZ-based Gateways.

TABLE 1–2 Secure Remote Access Features and Benefits *(Continued)*

Feature	Description	Benefit
Secure Remote Access core	Users achieve remote access through four components: ▪ Gateway ▪ NetFile ▪ Netlet ▪ Proxylet	This component has four parts: ▪ Gateway—Controls communication between the Portal Server and the various Gateway instances. ▪ NetFile—Enables remote access and operation of file systems and directories. ▪ Netlet—Ensures secure communication between the Netlet applet on the client browser, the Gateway, and the application servers. ▪ Proxylet—Proxylet sets itself up as a proxy server running on the client's machine, and modifies the proxy settings of the browser to point to itself (also referred to as the local proxy server). The local proxy server (Proxylet) then proxies all the intranet traffic through the gateway.
Universal access	Enables web browser based universal access with no client software installation or maintenance necessary.	Simplifies the IT administration and maintenance overhead while dramatically reducing the time and cost of deployment.
Netlet Proxy	Provides an optional component that extends the secure tunnel from the client, through the Gateway to the Netlet Proxy that resides in the intranet.	Restricts the number of open ports in a firewall between the demilitarized zone (DMZ) and the intranet.
Rewriter Proxy	Redirects HTTP requests to the Rewriter Proxy instead of directly to the destination host. The Rewriter Proxy in turn sends the request to the destination server.	Enables secure HTTP traffic between the Gateway and intranet computers and offers two advantages: ▪ If a firewall exists between the Gateway and server, the firewall needs to open only two ports: one between the Gateway and the Rewriter Proxy, and another between the Gateway and the Portal Server. ▪ HTTP traffic is secure between the Gateway and the intranet even if the destination server only supports HTTP protocol (no HTTPS).

Search Service

The Sun Java System Portal Server provides a Search Service the retrieves and categorizes information for users. The Search Service is used in the following channels:

- Subscription channel to summarize the number of hits (relevant information) that match each profile entry defined by the user for categorized documents and discussions.
- Discussion channel to individually search contents and rate the importance for comments.

Table 1–3 lists the search features and their benefits.

TABLE 1–3 Search Features and Benefits

Feature	Description	Benefit
Search Service	Enables the retrieval of documents based on criteria specified by the end user.	Saves users time by providing access to content.
Categorization	Organizes documents into a hierarchy. This categorization is often referred to as taxonomy.	Provides a different view of documents that enables browsing and retrieval.
Robot	The Search Service robot is an agent that crawls and indexes information across your intranet or the Internet.	Automatically searches and extracts links to resources, describes those resources, and puts the descriptions in the Search database (also called generation or indexing).
Discussions	A forum for multiple threaded discussions.	Contents are individually searchable and importance rating are given for of all comments
Subscriptions	Enables the user to track new or changed material in different areas of interest.	Discussions, search categories, and free-form searches (saved searches) can be tracked.

Content Personalizing

Personalization is the ability to deliver content based on selective criteria and offer services to a user.

Table 1–4 shows the personalization features and their benefits.

TABLE 1–4 Personalization Features and Benefits

Feature	Description	Benefit
Deliver content based on user's role	Portal Server includes the ability to automatically choose which applications users are able to access or to use, based on their role within the organization.	Increases employee productivity, improves customer relationships, and streamlines business relationships by providing quick and personalized access to content and services.
Enable users to customize content	Portal Server enables end users to choose what content they are interested in seeing. For example, users of a personal finance portal choose the stock quotes they would like to see when viewing their financial portfolio.	The information available in a portal is personalized for each individual. In addition, users can then customize this information further to their individual tastes. A portal puts control of the web experience in the hands of the people using the web, not the web site builders.
Aggregate and personalize content for multiple users	Portal Server enables an enterprise or service provider to aggregate and deliver personalized content to multiple communities of users simultaneously.	This enables a company to deploy multiple portals to multiple audiences from one product and manage them from a central management console. Also, new content and services can be added and delivered on demand without the need to restart Portal Server. All of this saves time and money, and ensures consistency in an IT organization.
Portal Communities	A Portal Community is an association of members and services that is created and administered by end-users.	The Portal Communities feature makes collaboration more accessible to end users by providing a way for end users to create their own Portal. End users are able to assign membership roles and choose which Portal services available to members.

Aggregation and Integration

One of the most important aspects of a portal is its ability to aggregate and integrate information, such as applications, services, and content. This functionality includes the ability to embed non-persistent information, such as stock quotes, through the portal, and to run applications within, or deliver them through, a portal.

Table 1–5 shows the aggregation and integration features and their benefits.

TABLE 1–5 Aggregation Features and Benefits

Feature	Description	Benefit
Aggregated information	The Portal Desktop provides the primary end-user interface for Portal Server and a mechanism for extensible content aggregation through the Provider Application Programming Interface (PAPI). The Portal Desktop includes a variety of providers that enable container hierarchy and the basic building blocks for building some types of channels.	Users no longer have to search for the information. Instead, the information finds them.
Consistent set of tools	Users can use the provided set of tools such as web-based email and calendaring software that follows them through their entire time at the company.	Users do not have to use one tool for one project, another tool for another location. Because these tools all work within the portal framework, the tools have a consistent look and feel and work similarly, reducing training time.
Collaboration	Portal Server provides control and access to data as a company-wide resource.	In many companies, data is seen as being owned by individual departments, instead of as a company-wide resource. The portal can act as a catalyst for making the data available in a controlled way to the people who need to use it. This broader, more immediate access can improve collaboration.
Integration	Portal Server enables you to use the Portal Desktop as the sole place for users to gain access to or launch applications and access data.	Integration with existing email, calendar, legacy, or web applications enables the portal to serve as a unified access point, enabling access to the information quickly and easily.

Using The Solution Life Cycle

The Solution lifecycle is a useful tool for keeping a deployment project on track. This guide uses these steps in presenting the information. The steps are the following:

- Business Analysis
- Technical Requirements
- Logical Design
- Deployment Design
- Deployment Implementation
- Operations

The following figure depicts the steps in the planning, design, and implementation of enterprise software solution based on the Java Enterprise System. Each enterprise has its own set of goals, requirements, and priorities to consider. Successful planning begins with analyzing the goals of an enterprise and determining the business requirements to meet those goals. The business requirements must then be converted into technical requirements that can be used as a basis for designing and implementing a system that can meet the goals of the enterprise.

For information see the *Sun Java Enterprise System 5 Deployment Planning Guide*.

Solution Life Cycle

The solution life cycle shown in the following figure depicts the steps in the planning, design, and implementation of an enterprise software solution based on Java Enterprise System. The life cycle is a useful tool for keeping a deployment project on track.

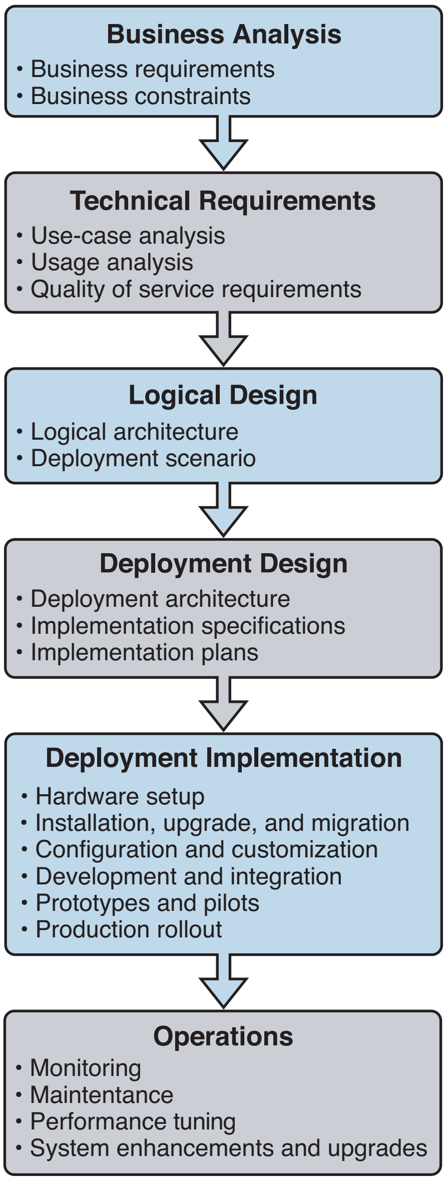


FIGURE 1–1 Solution Life Cycle

CHAPTER 2

Understanding Your Business Goals

During the business analysis phase of the solution life cycle you define business goals by analyzing a business problem and identifying the business requirements and business constraints to meet that goal.

This chapter contains the following sections:

- "About Business Analysis" on page 31
- "Defining Business Requirements" on page 32
- "Understanding User Behaviors and Patterns" on page 33

About Business Analysis

Business analysis starts with stating the business goals. You then analyze the business problems you must solve and identify the business requirements that must be met to achieve the business goals. Consider also any business constraints that limit your ability to achieve the goals. The analysis of business requirements and constraints results in a set of business requirements documents.

You use the resulting set of business requirements documents as a basis for deriving technical requirements in the technical requirements phase. Throughout the solution life cycle, you measure the success of your deployment planning and ultimately the success of your solution according to the analysis performed in the business analysis phase.

Defining Business Requirements

The business requirements of your portal affect deployment decisions. Understand your requirements to make correct assumptions that affect the accuracy of your deployment estimates.

The reasons you are offering your portal have a direct affect on how you implement your portal. You must define target population, performance standards, and other factors related to your goals.

Use these questions to help you identify the goals of your portal:

- What is your portal's biggest priority?
- What applications will the portal deliver?
- What is your target population?
- What performance standard is necessary?
- What transaction volume do you expect? What transaction volume do you expect during peak use?
- What response time is acceptable during peak use?
- What is the necessary level of concurrency? Concurrency is the number of users who can be connected at any given time?
- Should access to the portal be through the intranet or the Internet?
- Will your portal be deployed in one phase, or many phases?
- What are the business requirements of this portal? For example, do you want to enhance customer service? Increase employee productivity? Reduce the cost of doing business?
- What kind of portal do you need? For example, business-to-business, business-to-consumer, business-to-enterprise, or a hybrid?
- Who is your target audience?
- What services or functions will the portal deliver to users?
- How will the target audience benefit from the portal?
- What are the priorities for the portal?

(Optional) Use these questions to help identify your business objectives if you are deploying a secure portal:

- Do you need to increase employee productivity (by making your intranet applications and servers accessible over the Internet)?
- Do you need to provide secure access to your portal?
- Do you need to reduce the cost of ownership of an existing Virtual Private Network (VPN) solution?

- Do you want employees to access intranet applications such as Citrix and pcAnywhere from the Internet?
- Do you want your employees to explore intranet servers or machines from the Internet?
- Who is your target audience (all portal users, employees, or customers)?

Understanding User Behaviors and Patterns

Factors such as when users will use the portal and how users have used predecessor systems are keys to identifying your requirements. Study the people who will use your portal. If your organization's experience cannot provide these patterns, you can study the experience of other organizations and estimate them.

Use these questions to help you understand users:

- How many end users will you have? What is the size of your target audience?
- Will users login to the portal at the same time each day? Will they use the portal at work or somewhere else?
- Are users in the same time zone or in different time zones?
- How long do you expect the typical user to be connected, or have a valid portal session open? What use statistics do you have for existing applications? Do you have web traffic analysis figures for an existing portal?
- How many visitor sessions, or number of single-visitor visits, are likely within a predefined period of time?
- Is portal use likely to increase over time? Or stay stable?
- How fast will your user base grow?
- How have your users used an application that the portal will deliver to them?
- What portal channels do you expect users to use regularly?
- Will users set up and participate in communities? If so, what will team behavior and use of the portal be like?
- What expectations about your portal content do your users have? How have users used predecessor web-based information or other resources that your portal will offer?

Facilitating Productivity

Facilitating productivity requires that you examine the ways that individuals, teams, and your organization as a whole are productive, especially if your portal will provide communities. Consider teams that are set up for the entire organization as well as for specific parts of the organization and for short-term or long-term projects.

- How do individuals organize their daily work environments?
- How do new employees become productive quickly?
- How do new services become known?
- How do they become popular?
- How do employees find new documents?
- How do employees work with others?
- How do individuals work with others?
- How do individuals create ad-hoc teams?
- How do members of teams communicate?
- How do teams archive their ad-hoc or temporary efforts?
- How do you target new services to the correct groups and individuals?
- How do you measure the success of new services?
- How do you define and monitor individual productivity? Team productivity? Organizational productivity?
- How do you help improve individual productivity? Team productivity? Organizational productivity?

◆ ◆ ◆ CHAPTER 3

Identifying Your Technical Requirements

During the technical requirements phase of the solution life cycle you perform a usage analysis, identify use cases, and determine quality of service requirements for the proposed deployment solution.

This chapter contains the following sections:

- "About Technical Requirements" on page 35
- "Usage Analysis for Portal Server" on page 36
- "Usage Analysis for Secure Remote Access" on page 42
- "Portal Server Use Cases" on page 45
- "Quality of Service Requirements" on page 47
- "Designing for Localization" on page 51

About Technical Requirements

Technical requirements analysis begins with the business requirements documents created during the business analysis phase of the solution life cycle. Using the business analysis as a basis, you do the following:

- Perform a usage analysis to aid in determining expected load conditions.
- Create use cases that model typical user interaction with the system.
- Create a set of quality of service requirements (QoS) that define how a deployed solution must perform in areas such as response time, availability, security, and others.

The quality of service requirements are derived from the usage analysis and the use cases, keeping in mind business requirements and constraints previously identified. The quality of service requirements are later paired with logical architectures in the logical design phase to form a deployment scenario. The deployment scenario is the main input to the deployment design phase of the solution life cycle.

As with business analysis, no simple formula for technical requirements analysis exists that generates the usage analysis, use cases, and system requirements. Technical requirements analysis requires an understanding of the business domain, business objectives, and the underlying system technology.

Usage Analysis for Portal Server

You need to establish baseline sizing figures that can be used in the logical and architecture and deployment design. Your technical representative can provide you with an automated sizing tool to calculate the estimated number of CPUs your Portal Server deployment requires.

Note – Sizing requirements for a secure portal deployment using Sun Java™ System Secure Remote Access software are covered in "Usage Analysis for Secure Remote Access" on page 42.

You need to gather the following metrics for input to the sizing tool:

- "Peak Numbers" on page 36
- "Average Time Between Page Requests" on page 37
- "Concurrent Users" on page 37
- "Average Session Time" on page 38
- "Search Service Factors" on page 39
- "Page Configuration" on page 39

Other performance metrics that affect the number of CPUs a Portal Server deployment requires, but are not used by the sizing tool, are:

- Portal Desktop Configuration
- Hardware and Applications
- Back-end Servers
- Transaction Time
- Workload Conditions

A discussion of the these performance factors follows.

Peak Numbers

Maximum number of concurrent sessions defines how many connected users a Portal Server deployment can handle.

To calculate the maximum number of concurrent sessions, use this formula:

```
maximum number of concurrent sessions =
expected percent of users online * user base
```

To identify the size of the user base or pool of potential users for an enterprise portal, here are some suggestions:

- Identify only users who are active. Do not include users who are, for example, away on vacation, or on leave.
- Use a finite figure for user base. For an anonymous portal, estimate this number conservatively.
- Study access logs.
- Identify the geographic locations of your user base.
- Remember what your business plan states regarding who your users are.

Average Time Between Page Requests

Average time between page requests is how often, on average, a user requests a page from the Portal Server. Pages could be the initial login page to the portal, or a web site or web pages accessed through the Portal Desktop. A page view is a single call for a single page of information no matter how many items are contained on the page.

Though web server logs record page requests, using the log to calculate the average time between requests on a user basis is not feasible. To calculate the average time between page requests, you would probably need a commercially available statistics tool, such as the WebLoad performance testing tool. You can then use this figure to determine the number of concurrent users.

To perform stress testing, you can use the SLAMD Distributed Load Generation Engine. For more information about SLAMD, see http://slamd2.dev.java.net/.

Note – Page requests more accurately measure web server traffic than “hits.” Every time any file is requested from the web server counts as a hit. A single page call can record many hits, as every item on the page is registered. For example, a page containing 10 graphic files records 11 “hits”—one for the HTML page itself and one for each of the 10 graphic files. For this reason, page requests gives a more accurate determination of web server traffic.

Concurrent Users

A *concurrent user* is one connected to a running web browser process and submitting requests to or receiving results of requests from Portal Server. The maximum number of concurrent users is the highest possible number of concurrent users within a predefined period of time. Calculate the maximum number of concurrent users after you calculate the maximum number of concurrent sessions. To calculate the maximum number of concurrent users, use this formula:

```
concurrent users = number of concurrent sessions / average time between hits
```

For example, consider an intranet Portal Server example of 50,000 users. The number of connected sessions under its peak loads is estimated to be 80% of its registered user base. On average, a user accesses the Portal Desktop once every 10 minutes.

The calculation for this example is:

```
40000 / 10 = 4000
```

The maximum number of concurrent users during the peak hours for this Portal Server site should be 4,000.

Average Session Time

Average session time is the time between user login and logout averaged over a number of users. The length of the session time is inversely proportional to the number of logins occurring (that is, the longer the session duration, the fewer logins per second are generated against Portal Server for the same concurrent users base). Session time is the time between user login and user logout.

How the user uses Portal Server often affects average session time. For example, a user session involving interactive applications typically has a longer session time than a user session involving information only.

Search Service Factors

If your portal site will offer a Search channel, you need to include sizing factors for the Search Engine in your sizing calculations. Search Engine sizing requirements depend on the following factors:

- The size of index partitions on the active list of the index directory

  Partition size is directly proportional to the size and number of indexed and searchable terms.
- Average disk space requirement of a resource description (RD)

  To calculate this, use this formula:

  ```
  average disk space requirement =
  database size / number of RDs in database
  ```

  The average size adjusts for variations in sizes of RDs. A collection of long, complex RDs with many indexed terms and a list of short RDs with a few indexed terms require different search times, even if the complex RDs have the same number of RDs.

  RDs are stored in a hierarchical database format, where the intrinsic size of the database must be accounted for, even when no RD is stored.
- The number of concurrent users who perform search-related activities

  To calculate this, use this formula:

  ```
  number of concurrent users / average time between search hits
  ```

  Use the number of concurrent users value calculated in "Concurrent Users" on page 37.
- The type of search operators used

  Types of search functions include basic, combining, proximity, passage and field operator, and wildcard scans. Each function uses different search algorithms and data structures. Because differences in search algorithms and data structures increase as the number of search and indexed terms increase, the type of search function affects times for search result return trips.

Page Configuration

If you are using an authenticated portal, you must specify both Login Type and Desktop Type in the page configuration section of the automated sizing tool.

- **Login Type**. Describes the type of portal page (content configuration and delivery method) that end users initially see after submitting user name and password. This process is typically taxing on the system because the process involves checking credentials, initializing the session, and delivering initial content.

The Measured CPU Performance characteristic associated with the Login Type is the *Initial Desktop Display* variable.

- **Desktop Type**. Describes the type of portal pages (content configuration and delivery method) that end users see after the initial portal page. These pages are displayed with each subsequent interaction with the portal, or on Desktop refresh. Because the session has already been established and cached content can be exploited, less system resources are typically required and the pages are delivered more rapidly.

  The Measured CPU Performance characteristic associated with the Desktop Type is the *Desktop Reload* variable.

For both Login Type and Desktop Type, select the appropriate content configuration:

- Light-JSP. Describes a configuration of two tabs with five channels each.
- Regular-JSP. Describes a configuration of two tabs with seven channels each.
- Heavy-JSP. Describes a configuration of three tabs with seventeen channels each.

Tip – You can now give the above figures to your technical representative and ask that the sizing tool be run to identify your estimated number of CPUs.

Portal Desktop Configuration

Portal Desktop configuration explicitly determines the amount of data held in memory on a per-session basis.

The more channels on the Portal Desktop, the bigger data session size, and the less the throughput of Portal Server.

Another factor is how much interactivity the Portal Desktop offers. For example, channel clicks can generate load on Portal Server or on some other external server. If channel selections generate load on Portal Server, a higher user activity profile and higher CPU overhead occur on the node that hosts the Portal Desktop than on a node that hosts some other external server.

Hardware and Applications

CPU speed and size of the virtual machine for the Java platform (Java Virtual Machine or JVM™ software) memory heap affect Portal Server performance.

The faster the CPU speed, the higher the throughput. The JVM memory heap size, along with the heap generations tuning parameters, can also affect Portal Server performance.

Back-End Servers

Portal Server aggregates content from external sources. If external content providers cannot sustain the necessary bandwidth for Portal Server to operate at full speed, Portal Desktop rendering and throughput request times will not be optimum. The Portal Desktop waits until all channels are completed (or timed out) before it returns the request response to the browser.

Plan your back-end infrastructure carefully when you use channels that do the following:

- Scrape their content from external sources
- Access corporate databases, which typically have slow response times
- Provide email content
- Provide calendar content

Transaction Time

Transaction time, which is the delay taken for an HTTP or HTTPS operation to complete, aggregates send time, processing time, and response time figures.

You must plan for factors that can affect transaction time. These include:

- Network speed and latency. You need to examine latency over a Wide Area Network (WAN). Latency can significantly increase retrieval times for large amounts of data.
- The complexity of the Portal Desktop.
- The browser's connection speed.

  For example, a response time delay is longer with a connection speed of 33.6 kilobytes per second than with a LAN connection speed. However, processing time should remain constant. Transaction time through a dial-up connection should be faster than transaction time displayed by a load generation tool because it performs data compression.

When you calculate transaction time, size your Portal Server so that processing time under regular or peak load conditions does not exceed your performance requirement threshold and so that you can sustain processing time over time.

Workload Conditions

Workload conditions are the most predominantly used system and JVM software resources on a system. These conditions largely depend on user behavior and the type of portal you deploy.

The most commonly encountered workload conditions on Portal Server software affect:

- System performance

  Portal Server performance is impacted when a large number of concurrent requests are handled (such as a high activity profile). For example, during peak hours in a business-to-enterprise portal, a significant number of company employees connect to the portal at the same time. Such a scenario creates a CPU-intensive workload. In addition, the ratio of concurrent users to connected users is high.
- System capacity

  Portal Server capacity begins to be impacted when large numbers of users log in. As more users login, users use more of the available memory, and subsequently, less memory is available to process requests made to the server. For example, in a business-to-consumer web portal, a large number of logged-in users are redirected to external web sites once the initial Portal Desktop display is loaded. However, as more users continue to login, users create the need for more memory, even though the ratio of users submitting requests to Portal Server and the users merely logged-in is low.

  Depending on the user's behavior at certain times of the day, week, or month, Portal Server can switch between CPU-intensive and memory-intensive workloads. The portal site administrator must determine the most important workload conditions to size and tune the site to meet the enterprise's business goals.

Usage Analysis for Secure Remote Access

Use this section only if your organization is implementing a secure portal by installing Secure Remote Access. As you did for portal, for Secure Remote Access, you must first establish your Gateway instances baseline sizing estimate. A single machine can have one Gateway installation but multiple instances. Secure Remote Access enables you to install multiple Gateways, each running multiple instances.) Your design decisions help you make accurate estimates regarding Secure Remote Access user sessions and concurrency.

You must first establish your Gateway instances baseline sizing estimate. This baseline figure represents what you must have to satisfy your Gateway user sessions and concurrency needs.

Establishing an appropriate sizing estimate for your Secure Remote Access deployment is an iterative process. You might wish to change the inputs to generate a range of sizing results. Test these results against your original requirements. You can avoid most performance problems by formulating your requirements correctly and setting realistic expectations of Secure Remote Access performance.

Note – Properly sizing the Gateway is difficult, and using the Gateway sizing tool is only the beginning. Gateway performance depends more on throughput then on the number of users, active users, or user sessions. Any sizing information for the Gateway has to be based on a set of assumptions.

Scalability

You can choose between one, two, and four CPUs per Gateway instance. The number of CPUs bound to a Gateway instance determines the number of Gateway instances required for the deployment.

Using Secure Remote Access Prototype Numbers

If you have numbers from a prototype of the portal with Secure Remote Access, you can use these numbers in the Gateway sizing tool to arrive at more accurate results. You would fill in the following:

- Measured CPU Performance. The values used to help calculate the number of Gateway instances include:
  - Initial Portal Desktop Display, hits per second per CPU
  - Portal Desktop Reloads, hits per second per CPU
- Netlet Applications Block Size. This value specifies the Netlet application byte size. The Netlet dynamically determines the block size based on the application that is used. Block size determined by Netlet for a Telnet is based on the amount of data transferred.

Note – You do not need to specify the Page Configuration and Scalability options if you are using trial deployment numbers.

Key Performance Factors

Key performance factors are metrics that your technical representative uses as input to an automated sizing tool. The sizing tool calculates the estimated number of Gateway instances your Secure Remote Access deployment requires.

Identifying these key performance factors and giving them to your technical representative is the first step in formulating your baseline sizing figure.

These are the key performance factors:

- "Session Characteristics" on page 44
- "Netlet Usage Characteristics" on page 44

Note – After you calculate these key performance factors, give the figures to your technical representative. Ask that the Gateway sizing tool be run to identify the estimated number of Gateway instances.

Session Characteristics

The session characteristics of the Gateway include:

- Total number of Secure Remote Access (Gateway) users

  This represents the size of your user base or pool of potential users for the secure portal..
- Expected percentage of total users using the Gateway (at maximum load)

  Apply a percentage to your total number of users to determine this figure.
- Average time between page hits

  This is how often on average a user requests a page from the portal server.
- Session average time

  This determines how many logins per second that the Gateway must sustain for a given number of concurrent users.

Netlet Usage Characteristics

Consider the following Netlet characteristics of the Gateway, which can have a impact on calculating the number of Gateway instances:

- Netlet is enabled in the Portal Server administration console.

  If Netlet is enabled, the Gateway needs to determine whether the incoming traffic is Netlet traffic or Portal Server traffic. Disabling Netlet reduces this overhead since the Gateway assumes that all incoming traffic is either HTTP or HTTPS traffic. Disable Netlet only if you are sure you do not want to use any remote applications with Portal Server.
- Expected percentage of total users using Netlet

  Apply a percentage to your total number of users to determine this figure.
- Expected throughput

  Determine the expected throughput of your Gateway, expressed in kilobits per second (Kbps).
- Netlet Cipher (encryption) being used

  Choices include Native VM and Java software plugin ciphers.

Portal Server Use Cases

Use cases are written scenarios used to test and present the system's capabilities and form an important part of your high-level design. Though you implement use case scenarios toward the end of the project, formulate them early in the project once you have established your requirements.

When available, use cases can provide valuable insight into how the system is to be tested. Use cases are beneficial in identifying how you need to design the user interface from a navigational perspective. When designing use cases, compare them to your requirements to get a thorough view of their completeness and how you are to interpret the test results.

Use cases provide a method for organizing your requirements. Instead of a bulleted list of requirements, you organize them in a way that tells a story of how someone can use the system. This provides for greater completeness and consistency, and also gives you a better understanding of the importance of a requirement from a user perspective.

Use cases help to identify and clarify the functional requirements of the portal. Use cases capture all the different ways a portal would be used, including the set of interactions between the user and the portal as well as the services, tasks, and functions the portal is required to perform.

A use case defines a goal-oriented set of interactions between external actors and the portal system. (Actors are parties outside the system that interact with the system, and can be a class of users, roles users can play, or other systems.)

Use case steps are written in an easy-to-understand structured narrative using the vocabulary of the domain.

Use case scenarios are an instance of a use case, representing a single path through the use case. Thus, there may be a scenario for the main flow through the use case and other scenarios for each possible variation of flow through the use case (for example, representing each option).

Preparing to Develop Portal Server Use Cases

When developing use cases for your portal, keep the following elements in mind:

- **Priority**. Describes the priority, or ranking of the use case. For example, this could range from high to medium to low.
- **Context of use**. Describes the setting or environment in which the use case occurs.
- **Scope**. Describes the conditions and limitations of the use case.
- **Primary user**. Describes what kind of user this applies to, for example, an end user or an administrator.
- **Special requirements**. Describes any other conditions that apply.

- **Stakeholders**. Describes the people who have a vested interest in how a product decision is made or carried out.
- **Precondition**. Describes the prerequisites that must be met for the use case to occur.
- **Minimal guarantees**. Describes the minimum that must occur if the use case is not successfully completed.
- **Success guarantees**. Describes what happens if the use case is successfully completed.
- **Trigger**. Describes the particular item in the system that causes the event to occur.
- **Description**. Provides a step-by-step account of the use case, from start to finish.

Example Use Case: Authenticate Portal User

Table 3–1 describes a use case for a portal user to authenticate with the portal.

TABLE 3–1 Use Case: Authenticate Portal User

Item	Description
Priority	Must have.
Context of Use	Only authenticated end-users are allowed to gain access to the portal resources. This access restriction applies to all portal resources, including content and services. This portal relies on the user IDs maintained in the corporate LDAP directory.
Scope	The portal end-users identify themselves only once for a complete online session. In the case that an idle time-out occurs, the users must reidentify themselves. If the portal end-user identification fails more often than a specified amount of allowed retries, access to the intranet should be revoked or limited (deactivated) until a system administrator reactivates the account. In this case, the portal end-user should be advised to contact the authorized person. The identified portal end-users are able to access only the data and information that they are authorized for.
Primary User	Portal end-user.
Special Requirements	None.
Stakeholders	Portal end-user.

TABLE 3–1 Use Case: Authenticate Portal User *(Continued)*

Item	Description
Preconditions	The portal end-user: ■ is an authorized user. ■ has a standard corporate LDAP user ID. The LDAP user ID must be provided to each employee. ■ has an authorized LDAP entry. ■ has access to the corporate intranet. ■ does not have a guest account.
Minimal Guarantees	Friendly customer-centric message. Status—with error message indicating whom to call.
Success Guarantees	Presented with Portal Desktop home page. Authentication. Entitlement. Personal information.
Trigger	When any portal page is accessed and the end-user is not yet logged in.
Description	1. End-user enters the portal URL. 2. If the customization parameter [remember login] is set, then automatically login the user and provide a session ID. 3. If first time user, prompt for LDAP user ID and password. 4. End-user enters previously assigned user ID and password. 5. Information is passed to Access Manager for validation. 6. If authentication passes, assign session ID and continue. 7. If authentication fails, display error message, return end-user to login page; decrement remaining attempts; if preset attempts exceed limit, notify user and lock out the account.

Quality of Service Requirements

To provide a quality of service, address the following topics:

- "Performance" on page 48
- "Availability" on page 48
- "Scalability" on page 48
- "Security" on page 49

Performance

Work with portal system administrators and portal developers to setthe portal performance objectives based on the projected requirements of your portal. Objectives include the number of users, the number of concurrent users at peak load time and their usage pattern in accessing Portal Server.

You need to determine these two factors:

- Are you tuning for portal applications rapid response?
- Are you tuning for a large number of user concurrency?

As the number of users concurrently connected to the portal increase, the response time decreases given the same hardware and same set of parameters. Hence, gather information about the level of usage expected on your Portal Server, the anticipated number of concurrent users at any given time, the number of Portal Desktop activity requests, the amount of portal channel usage, acceptable response time for the end-user which is determined by your organization, and an optimal hardware configuration to meet the criteria.

Availability

Making Portal Server highly available involves ensuring high availability on each of the following components:

- **Portal Server**. In open mode, you can use a load balancer to detect a failed server component and redirect requests to other servers. In secure mode, Gateway components can detect the presence of a failed server component and redirect requests to other servers. (This is valid as long as the web container is the Sun Java System Web Server.)
- **Gateway**. A load balancer used with the Gateway detects a failed Gateway component and routes new requests to other Gateways. A load balancer also has the ability to intelligently distribute the workload across the server pool. Routing is restored when the failed Gateway recovers. Gateway components are stateless (session information is stored on the client in an HTTP cookie) so rerouting around a failed Gateway is transparent to users.
- **Directory Server**. A number of options make the LDAP directory highly available.
- **Netlet and Rewriter Proxies**In the case of a software crash, a watchdog process automatically restarts the proxies. In addition, the Gateway performs load balancing and failure detection failover for the proxies.

Scalability

Scalability is a system's ability to accommodate a growing user population, without performance degradation, by the addition of processing resources. The two general means of scaling a system are vertical and horizontal scaling. The subject of this section is the application of scaling techniques for the Portal Server.

Benefits of scalable systems include:

- Improved response time
- Fault tolerance
- Manageability
- Expendability
- Simplified application development
- Building modules

Vertical Scaling

In *vertical scaling*, CPUs, memory, multiple instances of Portal Server, or other resources are added to one machine. This enables more process instances to run simultaneously. In Portal Server, you want to make use of this by planning and sizing to the number of CPUs you need.

Horizontal Scaling

In *horizontal scaling*, machines are added. This also enables multiple simultaneous processing and a distributed work load. In Portal Server, you make use of horizontal scaling because you can run the Portal Server, Directory Server and Access Manager on different nodes. Horizontal scaling can also make use of vertical scaling, by adding more CPUs, for example.

Additionally, you can scale a Portal Server installation horizontally by installing server component instances on multiple machines. Each installed server component instance executes an HTTP process, which listens on a TCP/IP port whose number is determined at installation time. Gateway components use a round-robin algorithm to assign new session requests to server instances. While a session is established, an HTTP cookie, stored on the client, indicates the session server. All subsequent requests go to that server.

The section "Designing for Localization" on page 51, discusses an approach to a specific type of configuration that provides optimum performance and horizontal scalability.

Security

Security is the set of hardware, software, practices, and technologies that protect a server and its users from malicious outsiders. In that regard, security protects against unexpected behavior.

You need to address security globally and include people and processes as well as products and technologies. Unfortunately, too many organizations rely solely on firewall technology as their only security strategy. These organizations do not realize that many attacks come from employees, not outsiders. Therefore, you need to consider additional tools and processes when creating a secure portal environment.

Operating Portal Server in a secure environment involves making certain changes to the Solaris™ Operating Environment, the Gateway and server configuration, the installation of firewalls, and user authentication through Directory Server and SSO through Access Manager. In addition, you can use certificates, SSL encryption, and group and domain access.

Access Control

Portal Server relies on the HTTPS encryption protocol, in addition to UNIX system security, for protecting the Portal Server system software.

Security is provided by the web container, which you can configure to use SSL, if desired. Portal Server also supports SSL for authentication and end-user registration. By enabling SSL certificates on the web server, the Portal Desktop and other web applications can also be accessed securely. You can use the Access Manager policy to enforce URL-based access policy.

Portal Server depends on the authentication service provided by Access Manager and supports single sign-on (SSO) with any product that also uses the Access Manager SSO mechanism. The SSO mechanism uses encoded cookies to maintain session state.

Another layer of security is provided by Secure Remote Access. It uses HTTPS by default for connecting the client browser to the intranet. The Gateway uses the Rewriter or Proxylet to enable all intranet web sites to be accessed without exposing them directly to the Internet. The Gateway also provides URL-based access policy enforcement without having to modify the web servers being accessed.

Communication from the Gateway to the server and intranet resources can be HTTPS or HTTP. Communication within the Portal Server system, for example between web applications and the directory server, does not use encryption by default, but it can be configured to use SSL.

UNIX User Installation

You can install and configure Portal Server to run under three different UNIX users:

- `root`. This is the default option. All Portal Server components are installed and configured to run as the system superuser. Some security implications arise from this configuration:
  - An application bug can be exploited to gain root access to the system.
  - You need root access to modify some of the templates. Root access raises potential security concerns as this responsibility is typically delegated to non-system administrators.
- **User** `nobody`. You can install Portal Server as the user `nobody` (uid 60001). This can improve the security of the system, because the user `nobody` does not have any privileges and cannot create, read, or modify the system files. This feature prevents user `nobody` from using Portal Server to gain access to system files and break into the system.

  The user `nobody` does not have a password, which prevents a regular user from becoming nobody. Only the superuser can change users without being prompted for a password. Thus, you still need `root` access to start and stop Portal Server services.
- **Non-root user**. You can run Portal Server as a regular UNIX user. The security benefits of a regular user are similar to the security benefits provided by the user `nobody`. A regular UNIX user has additional benefits as this type of user can start, stop, and configure services. After installation, you need to change ownership of some files.

See the *Sun Java Enterprise System 5 Installation Guide* for more information.

Limiting Access Control

While the traditional security UNIX model is typically viewed as all-or-nothing, you can use alternative tools to provide some additional flexibility. These tools provide the mechanisms needed to create a fine grain access control to individual resources, such as different UNIX commands. For example, this toolset enables Portal Server to be run as `root`, while allowing certain users and roles superuser privileges to start, stop, and maintain the Portal Server framework.

These tools include:

- **Role-Based Access Control (RBAC)**. Solaris 8 and higher include the Role-Based Access Control (RBAC) to package superuser privileges and assign them to user accounts. RBAC enables separation of powers, controlled delegation of privileged operations to users, and a variable degree of access control.
- **Sudo**. Sudo is publicly available software that enables a system administrator to give certain users the ability to execute a command as another user. See:

  `http://www.courtesan.com/sudo/sudo.html`

Using a Secure Access Zone

For maximum security, the Gateway is installed between two firewalls. The outermost firewall enables only SSL traffic from the Internet to the Gateways, which then direct traffic to servers on the internal network.

Designing for Localization

Localization is the process of adapting text and cultural content to a specific audience. Localization can be approached in two different ways:

- Localization of the entire product into a language that we don't provide. This is usually done by a professional service organization.
- Localization of parts of Portal Server that can be translated to support localization include:
  - Template and JSP files
  - Resource Bundles
  - Display profile properties

For advanced language localization, create a well-defined directory structure for template directories. To preserve the upgrade path, maintain custom content and code outside of default directories. See the *Sun Java System Portal Server 7.1 Developer's Guide* for more information on localization.

CHAPTER 4

Designing Your Logical Architecture

During the Logical design phase of the solution lifecycle, you formulate a logical architecture. A logical architecture identifies the interrelationships of the software components needed to implement your solution.

This chapter contains the following sections:

- "Analysis of Logical Architecture" on page 53
- "Portal Server Components" on page 55
- "Secure Remote Access Components" on page 57
- "Portal Server Nodes" on page 67
- "Example Portal Server Logical Architectures" on page 71
- "Example Secure Remote Access Logical Architectures" on page 83
- "Deployment Scenario" on page 92

Analysis of Logical Architecture

A high-level logical architecture provides the basis for a low-level logical architecture. The high-level logical architecture needs to meet the business and technical needs that you previously established. The logical architecture is broken down according to the various applications that comprise the system as a whole and the way in which users interact with it. In general, the logical architecture includes Portal Server Secure Remote Access, high availability, security (including Access Manager, and Directory Server architectural components.

The high- and low-level architectures also need to account for any factors beyond the control of the portal, including your network, hardware failures, and improper channel design.

The low-level architecture specifies such items as the physical architecture, network infrastructure, Portal Desktop channel and container design and the actual hardware and software components.

High-Level Logical Architecture

The high-level logical architecture to supports both the business and technical requirements and addresses questions such as:

- Does the proposed architecture support both the business and technical requirements?
- Can any modifications strengthen this architecture?
- Are there alternative architectures that might accomplish this?
- What is the physical layout of the system?
- What is the mapping of various components and connectivity?
- What is the logical definition describing the different categories of users and the systems and applications users have access to?
- Does the design account for adding more hardware to the system as required by the increase in web traffic over time?

Low-Level Logical Architecture

Low-level architecture focuses on specifying the processes and standards you use to build your portal solution, and specifying the actual hardware and software components of the solution, including:

- The Portal Server complex of servers.
- Network connectivity, describing how the portal complex attaches to the "outside world." Within this topic, you need to take into account security issues, protocols, speeds, and connections to other applications or remote sites.
- Information architecture, including user interfaces, content presentation and organization, data sources, and feeds.
- Access Manager architecture, including the strategy and design of organizations, suborganizations, roles, groups, and users, which is critical to long-term success.
- Integration strategy, including how the portal acts as an integration point for consolidating and integrating various information, and bringing people together in new ways.

Portal Server Components

Portal Server deployment consists of the following components:

- Sun Java™ System Access Manager

  Access Manager provides user and service management, authentication and single sign-on services, policy management, logging service, debug utility, the administration console, and client support interfaces for Portal Server. This consists of:

  - Java Development Kit (JDK™)

    Java Development Kit software provides the Java run-time environment for all Java software in Portal Server and its underlying components. Portal Server depends on the JDK software in the web container.

  - Network Security Services for Java software
  - Sun Java System Web Server
  - Java API for XML Processing (JAXP)
- Sun Java System Directory Server

  Directory Server provides the primary configuration and user profile data repository for Portal Server. The Directory Server is LDAP compliant and implemented on an extensible, open schema.

- Web Containers
  - Sun Java System Web Server
  - Sun Java System Application Server Enterprise Edition

    The following web containers can be used in place of the Web Server and Application Server software:

  - BEA WebLogic Server
  - IBM WebSphere® Application Server

    See the *Sun Java Enterprise System 2006Q5 Installation Guide* for information on deploying Portal Server in various web containers.

Note – See the latest Portal Server Release Notes for specific versions of products supported by Portal Server.

In addition to the components that make up the portal, your design should include (but is not limited to) the following:

- Contents from RDBMs
- Third-party content providers
- Custom developed providers and content

- Integration with back-end systems such as messaging and calendaring systems
- Role of the Content Management System
- Customer Resource Management
- Whether the portal runs in open or secure mode (requires Secure Remote Access)
- Usage estimates, which include your assumptions on the total number of registered users, average percentage of registered users logged in per day, average concurrent users that are logged in per day, average login time, average number of content channels that a logged in user has selected, and average number of application channels that a logged in user has selected.

Additionally, you need to consider how the following three network zones fit into your design:

- **Internet.** The public Internet is any network outside of the intranet and DMZ. Users portal server and securely access the Gateway and from here.
- **Demilitarized Zone (DMZ).** A secure area between two firewalls, enabling access to internal resources while limiting potential for unauthorized entry. The Gateway resides here where it can securely direct traffic from the application and content servers to the Internet.
- **Intranet.** Contains all resource servers. This includes intranet applications, web content servers, and application servers. The Portal Server and Directory Server reside here.

The logical architecture also describes the Portal Desktop look and feel, including potential items such as:

- Default page, with its default banner, logo, channels; total page weight, that is, total number of bytes of all the components of the page, including HTML, style sheet, JavaScript™, and image files; total number of HTTP requests for the page, that is, how many HTTP requests are required to complete downloading the page.
- Personalized pages, with channels that users can conceivably display and what preferences are available.

The logical architecture is where you also develop a caching strategy, if your site requires one. If the pages returned to your users contain references to large numbers of images, Portal Server can deliver these images for all users. However, if these types of requests can be offloaded to a reverse proxy type of caching appliance, you can free system resources so that Portal Server can service additional users. Additionally, by placing a caching appliance closer to end users, these images can be delivered to end users somewhat more quickly, thus enhancing the overall end user experience.

Secure Remote Access Components

This section describes the following Secure Remote Access components:

- "Secure Remote Access Gateway" on page 57
- "Netlet" on page 60
- "Netlet Proxy" on page 62
- "NetFile" on page 62
- "Rewriter" on page 65
- "Rewriter Proxy" on page 66
- "Proxylet" on page 66

Secure Remote Access Gateway

The Secure Remote Access Gateway is a stand-alone Java process that can be considered to be stateless, since state information can be rebuilt transparently to the end user. The Gateway listens on configured ports to accept HTTP and HTTPS requests. Upon receiving a request, the Gateway checks session validity and header information to determine the type of request. Depending on the type of request, the Gateway performs the following:

- **Netlet request**. Routes the request (traffic) to the server specified in the Netlet rule that the user clicked in the Portal Desktop.
- **HTTP(S) traffic**. Routes the request to the server as specified by the HTTP header. Upon receiving a response from the server, the Gateway translates the response so that all intranet links within the response work on the extranet.

All the Gateway configuration information is stored in the Access Manager's LDAP database as a profile. A gateway profile consists of all the configuration information related to the Gateway except.

Machine-specific information, such as host name and IP address, is stored in a configuration file in the local file system where the Gateway is installed. This information enables one gateway profile to be shared between Gateways that are running on multiple machines.

As mentioned previously, you can configure the Gateway to run in both HTTP and HTTPS, simultaneously. This configuration helps both intranet and extranet users to access the same Gateway: extranet users over HTTPS, and intranet users over HTTP (without the overhead of SSL).

Multiple Gateway Instances

If desired, you can run multiple Gateway instances on a single machine. Each Gateway instance listens on separate ports. You can configure Gateway instances to contact the same Portal Server instance, or different Portal Server instances. When running multiple instances of a Gateway on the same machine, you can associate an independent certificate database with each instance of the Gateway, and bind that Gateway to a domain. This provides the flexibility of having a different Gateway server certificate for each domain.

Multiple Portal Server Instances

Note – Session stickiness is not required in front of a Gateway unless you are using Netlet. Performance is improved with session stickiness. On the other hand, session stickiness to the Portal Server instances is enforced by Secure Remote Access.

Proxies

The Gateway uses proxies that are specified in its profile to retrieve contents from various web servers within the intranet and extranet. You can dedicate proxies for hosts and DNS subdomains and domains. Depending on the proxy configuration, the Gateway uses the appropriate proxy to fetch the required contents. If the proxy requires authentication, the proxy name is stored as part of the gateway profile, that the Gateway uses automatically, when connecting to the proxy.

Gateway and HTTP Basic Authentication

The Gateway supports basic authentication, that is, prompting for a user ID and password but not protecting those credentials during transmission from the user's computer to the site's web server. Such protection usually requires the establishment of a secure HTTP connection, typically through the use of SSL.

If a web server requires basic authentication the client prompts for user name and password and sends the information back to the requesting server. With the Gateway enabled for HTTP basic authentication, it captures the user name and password information and stores a copy in the user's profile in the Access Manager for subsequent authentications and login attempts. The original data is passed by the Gateway to the destination web server for basic authentication. The web server performs the validation of the user name and password.

The Gateway also enables fine control of denying and allowing this capability on an individual host basis.

Gateway and SSL Support

The Gateway supports both SSL v2 and SSL v3 encryption while running in HTTPS mode. You can use the Portal Server administration console to enable or disable specific encryption. The Gateway also supports Transport Layer Security (TLS).

SSL v3 has two authentication modes.

- **Mandatory server authentication**. The client must authenticate the server.
- **Optional authentication**. The server is configured to authenticate the client.

Personal Digital Certificate (PDC) authentication is a mechanism that authenticates a user through SSL client authentication. The Gateway supports PDC authentication with the support of Access Manager authentication modules. With SSL client authentication, the SSL handshake ends at the Gateway. This PDC-based authentication is integrated along with the Access Manager's certificate-based authentication. Thus, the client certificate is handled by Access Manager and not by the Gateway.

If the session information is not found as part of the HTTP or HTTPS request, the Gateway directly takes the user to the authentication page by obtaining the login URL from Access Manager. Similarly, if the Gateway finds that the session is not valid as part of a request, it takes the user to the login URL and at successful login, takes the user to the requested destination.

After the SSL session has been established, the Gateway continues to receive the incoming requests, checks session validity, and then forwards the request to the destination web server.

The Gateway server handles all Netlet traffic. If an incoming client request is Netlet traffic, the Gateway checks for session validity, decrypts the traffic, and forwards it to the application server. If Netlet Proxy is enabled, the Gateway checks for session validity and forwards it to Netlet Proxy. The Netlet Proxy then decrypts and forwards it to the application server.

Note – Because 40-bit encryption is very insecure, the Gateway provides an option that enables you to reject connections from a 40-bit encryption browser.

Gateway Access Control

The Gateway enforces access control by using Allowed URLs and Denied URLs lists. Even when URL access is allowed, the Gateway checks the validly of the session against the Access Manager session server. URLs that are designated in the Non Authenticated URL list bypass session validation, as well as the Allowed and Denied lists. Entries in the Denied URLs list take precedence over entries in the Allowed URLs list. If a particular URL is not part of any list, then access is denied to that URL. The wildcard character, *, can also be used as a part of the URL in either the Allow or Deny list.

Gateway Logging

You can monitor the complete user behavior by enabling logging on the Gateway. The Gateway uses the Portal Server logging API for creating logs.

Using Accelerators with the Gateway

You can configure accelerators, which are dedicated hardware co-processors, to off-load the SSL functions from a server's CPU. Using accelerators frees the CPU to perform other tasks and increases the processing speed for SSL transactions.

Netlet

Netlet provides secure access to fixed port applications and some dynamic port applications that are available on the intranet from outside the intranet. The client can be behind a remote firewall and SSL proxy, or directly connected to the Internet. All the secure connections made from outside the intranet to the intranet applications through the Netlet are controlled by Netlet rules.

A Netlet applet running on the browser sets up an encrypted TCP/IP tunnel between the remote client machine and intranet applications on the remote hosts. Netlet listens to and accepts connections on preconfigured ports, and routes both incoming and outgoing traffic between the client and the destination server. Both incoming and outgoing traffic is encrypted using an encryption algorithm selected by the user, or configured by the administrator. The Netlet rule contains the details of all servers, ports, and encryption algorithms used in a connection. Administrators create Netlet rules by using the Portal Server administration console.

Static and Dynamic Port Applications

Static port applications run on known or static ports. Examples include IMAP and POP servers, Telnet daemons, and jCIFS. For static port applications, the Netlet rule includes the destination server port so that requests can be routed directly to their destinations.

Dynamic applications agree upon a port for communication as part of the handshake. You can include the destination server port as part of the Netlet rule. The Netlet needs to understand the protocol and examine the data to find the port being used between the client and the server. FTP is a dynamic port application. In FTP, the port for actual data transfer between the client and server is specified through the `PORT` command. In this case, the Netlet parses the traffic to obtain the data channel port dynamically.

Currently, FTP and Microsoft Exchange are the only dynamic port applications that Portal Server supports.

Note – Although Microsoft Exchange 2000 is supported with Netlet, the following constraints apply:

- For Portal Server versions before 6.3:
  - You must configure Exchange to use STATIC ports.
  - Netlet does not work with Windows 2000 and XP because Windows 2000 and XP clients reserve the Exchange port (port 135) for the RPC Portmapper, which Active Directory uses. Previous versions of Windows did not reserve this port. Because the port is reserved, you cannot assign Netlet to it, and thus the port cannot provide the necessary tunneling.
  - The Outlook 2000 client has the limitation that it does not enable you to change the port on which you want to connect to the Exchange server.
- For Portal Server 6.3 and later versions, Proxylet technology was introduced for use with OWA and Sun Java Enterprise Server, Portal Server Secure Remote Access deployments issues. Portal Server Administrators should consider this technology for a better user experience.

Netlet and Application Integration

Netlet works with many third parties such as Graphon, Citrix, and pcAnywhere. Each of these products provides secure access to the user's Portal Desktop from a remote machine using Netlet.

Split Tunneling

Split tunneling allows a VPN client to connect to both secure sites and non-secure sites, without having to connect or disconnect the VPN—in this case, the Netlet—connection. The client determines whether to send the information over the encrypted path, or to send it by using the non-encrypted path. The concern over split tunneling is that you could have a direct connection from the non-secure Internet to your VPN-secured network, via the client. Turning off split tunneling (not allowing both connections simultaneously) reduces the vulnerability of the VPN (or in the case of Netlet) connection to Internet intrusion.

Though Portal Server does not prohibit nor shut down multiple network connections while attached to the portal site, it does prevent unauthorized users from "piggybacking" on other users's sessions in the following ways:

- Netlet is an application specific VPN and not a general purpose IP router. Netlet only forwards packets that have been defined by a Netlet rule. This differs from the standard VPN approach that gives you complete LAN access once you've connected to the network.
- Only an authenticated portal user can run the Netlet. No portal application can be run until the user has been successfully authenticated, and no new connections can be made if an authenticated session does not exist.

- All access controls in place on the application side are still in effect so that an attacker would also have to break in to the back-end application.
- Every Netlet connection results in a dialog box posted by the Netlet (running in the authenticated user's JVM™) to the authenticated user's display. The dialog box asks for verification and acknowledgement to permit the new connection. For attackers to be able to utilize a Netlet connection, attackers would need to know that the Netlet was running, the port number it was listening on, how to break the back-end application, and convince the user to approve the connection.

Netlet Proxy

A Netlet Proxy helps reduce the number of open ports needed in the firewall to connect the Gateway and the destination hosts.

For example, consider a configuration where users need Netlet to connect with a large number of Telnet, FTP, and Microsoft Exchange servers within the intranet. Assume that the Gateway is in a DMZ. If it routes the traffic to all the destination servers, a large number of ports would need to be open in the second firewall. To alleviate this problem, you can use a Netlet Proxy behind the second firewall and configure the Gateway to forward the traffic to the Netlet Proxy. The Netlet Proxy then routes all the traffic to the destination servers in the intranet and you reduce the number of open ports required in the second firewall. You can also deploy multiple Netlet Proxies behind the second firewall to avoid a single point of failure.

You could also use a third-party proxy to use only one port in the second firewall.

Note – Installing the Netlet Proxy on a separate node can help with Portal Server response time by offloading Netlet traffic to a separate node.

NetFile

NetFile enables remote access and operation of file systems that reside within the corporate intranet in a secure manner.

NetFile uses standard protocols such as NFS, jCIFS, and FTP to connect to any of the UNIX or Windows file systems that are permissible for the user to access. NetFile enables most file operations that are typical to file manager applications.

Components

To provide access to various file systems, NetFile has three components:

- **NetFile Java 1 Applet**. Has an AWT-based user interface. For use with older browsers that cannot support Java 2.
- **NetFile Java 2 Applet**. Has a Swing-based user interface. For use with browsers that support Java plug-ins.
- **NetFile servlet(s)**. Two NetFile servlets are present in the web container, one for each kind of NetFile applet. The servlets are responsible for connecting to different types of file systems, carrying out the operations that NetFile is configured to handle, and sending the information back to the applets for display.

NetFile is internationalized and provides access to file systems irrespective of their locale (character encoding).

NetFile uses Access Manager to store its own profile, as well as user settings and preferences. You administer NetFile through the Portal Server administration console.

Initialization

When a user selects a NetFile link in the Portal Server Desktop, the NetFile servlet checks if the user has a valid SSO token and permission to execute NetFile. If so, the applet is rendered to the browser. The NetFile applet connects back to the servlet to get its own configuration such as size, locale, resource bundle, as well as user settings and preferences. NetFile obtains the locale information and other user information (such as user name, mail ID, and mail server) using the user's SSO token. The user settings include any settings that the user has inherited from an organization or role, settings that are customized by the user, and settings that the user has stored upon exit from a previous NetFile session.

Validating Credentials

NetFile uses the credentials supplied by users to authenticate users before granting access to the file systems.

The credentials include a user name, password, and Windows or Novell domain (wherever applicable). Each share can have an independent password, therefore, users need to enter their credentials for every share (except for common hosts) that you add.

NetFile uses UNIX Authentication from the Access Manager to grant access to NFS file systems. For file systems that are accessed over FTP and jCIFs protocols, NetFile uses the methods provided by the protocol itself to validate the credentials.

Access Control

NetFile provides various means of file system access control. You can deny access to users to a particular file system based on the protocol. For example, you can deny a particular user, role, or organization access to file systems that are accessible only over NFS.

You can configure NetFile to allow or deny access to file systems at any level, from organization, to suborganization, to user. You can also allow or deny access to specific servers. Access can be allowed or denied to file systems for users depending on the type of host, including Windows, FTP, NFS, and FTP over NetWare. For example, you can deny access for Windows hosts to all users of an organization. You can also specify a set of common hosts at an organization or role level, so that all users in that organization or role can access the common hosts without having to add them for each and every member of the organization or role.

As part of the NetFile service, you can configure the Allowed URLs or Denied URLs lists to allow or deny access to servers at the organization, role, or user level. The Denied URLs list takes precedence over the Allowed URLs. The Allowed URLs and Denied URLs lists can contain the * wildcard to allow or deny access to a set of servers under a single domain or subdomain.

Security

When you use NetFile with Secure Remote Access configured for SSL, all connections made from NetFile applets to the underlying file system happen over the SSL connection established between the Gateway and the browser. Because you typically install the Gateway in a DMZ, and open a limited number of ports (usually only one) in the second firewall, you do not compromise security while providing access to the file systems.

Special Operations

NetFile is much like a typical file manager application with a set of features that are appropriate for a remote file manager application. NetFile enables users to upload and download files between the local and remote file systems (shares). You can limit the size of the upload file (from the local to the remote file system) through the Portal Server administration console.

NetFile also enables users to select multiple files and compress them by using GZIP and ZIP compression. Users can select multiple files and send them in a single email as multiple attachments. NetFile also uses the SSO token of Access Manager to access the user's email settings (such as IMAP server, user name, password, and reply-to address) for sending email.

Double-clicking a file in the NetFile window launches the application corresponding to the MIME type and opens the file. NetFile provides a default MIME types configuration file that has mappings for most popular file types (extensions) and MIME-types that you can edit for adding new mappings.

You can search for files and display the list in a separate window using NetFile. The results of each search are displayed in a new window while maintaining the previous search result windows. The type of character encoding to be used for a particular share is user configurable, and is part of the share's setting. If no character encoding is specified, NetFile uses ISO-8859-1 while working with the shares. The ISO-8859-1 encoding is capable of handling most common languages. ISO-8859-1 encoding gives NetFile the capability to list files in any language and to transferring files in any language without damaging the file contents.

NetFile creates temporary files only when mailing files (in both NetFile Java 1 and Java 2). Temporary files are not created during uploading and downloading files between Windows file systems and the local file systems over the jCIFS protocol.

Note – NetFile supports deletion of directories and remote files. All the contents of remote directories are deleted recursively.

NetFile and Multithreading

NetFile uses multithreading to provide the flexibility of running multiple operations simultaneously. For example, users can launch a search operation, start uploading files, then send files by using email. NetFile performs all three operations simultaneously and still permit the user to browse through the file listing.

Rewriter

Rewriter is an independent component that translates all URIs (in both HTML and JavaScript code) to ensure that the intranet content is always fetched through the Gateway. You define a ruleset (a collection of rules) that identifies all URLs that need to be rewritten in a page. The ruleset is an XML fragment that is written according to a Document Type Definition (DTD). Using the generic ruleset that ships with the Rewriter, you can rewrite most URLs (but not all) without any additional rules. You can also associate rulesets with domains for domain-based translations.

An external ruleset identifies the URI in the content. Any request that needs to be served by Secure Remote Access follows this route:

▼ To Route Secure Remote Access Requests

1. **From the request, Secure Remote Access identifies the URI of the intranet page or Internet page that needs to be served.**
2. **Secure Remote Access uses the proxy settings to connect to the identified URI.**
3. **The domain of the URI is used to identify the ruleset to be used to rewrite this content.**
4. **After fetching the content and ruleset, Secure Remote Access inputs these to the Rewriter where identified URIs are translated.**
5. **The original URI is replaced with the rewritten URI.**
6. **This process is repeated until the end of the document is reached.**

7 The resultant Rewriter output is routed to the browser.

Rewriter Proxy

To minimize the number of open ports in the firewall, use the Rewriter Proxy. When you install the Rewriter Proxy, HTTP requests are redirected to the Rewriter Proxy instead of directly to the destination host. The Rewriter Proxy in turn sends the request to the destination server.

Using the Rewriter Proxy enables secure HTTP traffic between the Gateway and intranet computers and offers two advantages:

- If a firewall is between the Gateway and server, the firewall needs to open only two ports. One firewall is between the Gateway and the Rewriter Proxy and another is between the Gateway and the Portal Server.
- You can use a third-party proxy to use only one port in the second firewall to read the Rewriter Proxy.
- HTTP traffic is now secure between the Gateway and the intranet even if the destination server only supports HTTP protocol (not HTTPS).

Note – You can run multiple Rewriter Proxies to avoid a single point of failure and achieve load balancing.

Proxylet

Proxylet is a dynamic proxy server that runs on a client machine. Proxylet redirects a URL to the Gateway. It does this by reading and modifying the proxy settings of the browser on the client machine so that the settings point to the local proxy server or Proxylet.

It supports both HTTP and SSL, inheriting the transport mode from the Gateway. If the Gateway is configured to run on SSL, Proxylet establishes a secure channel between the client machine and the Gateway. Proxylet uses the Java 2 Enterprise Edition API if the client JVM is 1.4 or higher or if the required jar files reside on the client machine. Otherwise it uses the KSSL API.

Proxylet is enabled from the Portal Server administration console where the client IP address and port are specified.

Unlike Rewriter, Proxylet is an out-of-the-box solution with very little or no post-installation changes. Also Gateway performance improves because Proxylet does not deal with web content.

Portal Server Nodes

Usually, but not always, you deploy Portal Server software on the following different portal nodes (servers) that work together to implement the portal:

- **Portal Server node.** The web server where Portal Server resides. You can also install the Search component on this node if desired. Access Manager can reside here.
- **Access Manager node.**The server where Access Manager can reside. Access Manager does not have to reside on the same node as Portal Server.
- **Search node.** (Optional) The server you use for the Portal Server Search service. You can install the Portal Server Search service on its own server for performance, scalability and availability reasons.
- **Gateway nodes.** (Optional) The server where the Secure Remote Access Gateway resides. You can install the Gateway on the portal node. Because you locate the Gateway in the DMZ, the Gateway is installed on a separate, non-portal node.
- **Netlet Proxy node.** (Optional) The server used to run applications securely between users' remote desktops and the servers running applications on your intranet.
- **Rewriter Proxy node.** (Optional) The server used to run applications securely between users' remote desktops and the servers running applications on your intranet.
- **Directory Server node.** The server running Directory Server software. You can install Directory Server on a non-portal node.
- **Other servers.** These servers, such as mail, file, and legacy servers, provide backend support, data, and applications to portal users.

Portal Server and Access Manager on Different Nodes

Portal Server and Access Manager can be located on different nodes. This type of deployment provides the following advantages:

- Identity services can be deployed separately from portal services. Portal Server can be one of many applications using identity services.
- Authentication and policy services can be separate from provider applications including Portal Server related applications.
- Access Manager can be used by other web containers to assist with development of portal customizations.

Note – When Portal Server and Access Manager are on different nodes, the Access Manager SDK must reside on the same node as Portal Server. The web application and supporting authentication daemons can reside on a separate node from the Portal Server instance.

The Access Manager SDK consists of the following components:

- Identity Management SDK—provides the framework to create and manage users, roles, groups, containers, organizations, organizational units, and sub-organizations.
- Authentication API and SPI—provides remote access to the full capabilities of the Authentication Service.
- Utility API — manages system resources.
- Loggin API and SPI — records, among other things, access approvals, access denials and user activity.
- Client Detection API — detects the type of client browser that is attempting to access its resources and respond with the appropriately formatted pages.
- SSO API—provides interfaces for validating and managing session tokens, and for maintaining the user's authentication credentials.
- Policy API — evaluates and manages Access Manager policies and provides additional functionality for the Policy Service.
- SAML API — exchanges acts of authentication, authorization decisions and attribute information.
- Federation Management API — adds functionality based on the Liberty Alliance Project specifications.

Portal Server System Communication Links

Figure 4–1 shows the processes and communication links of a Portal Server system that are critical to the availability of the solution.

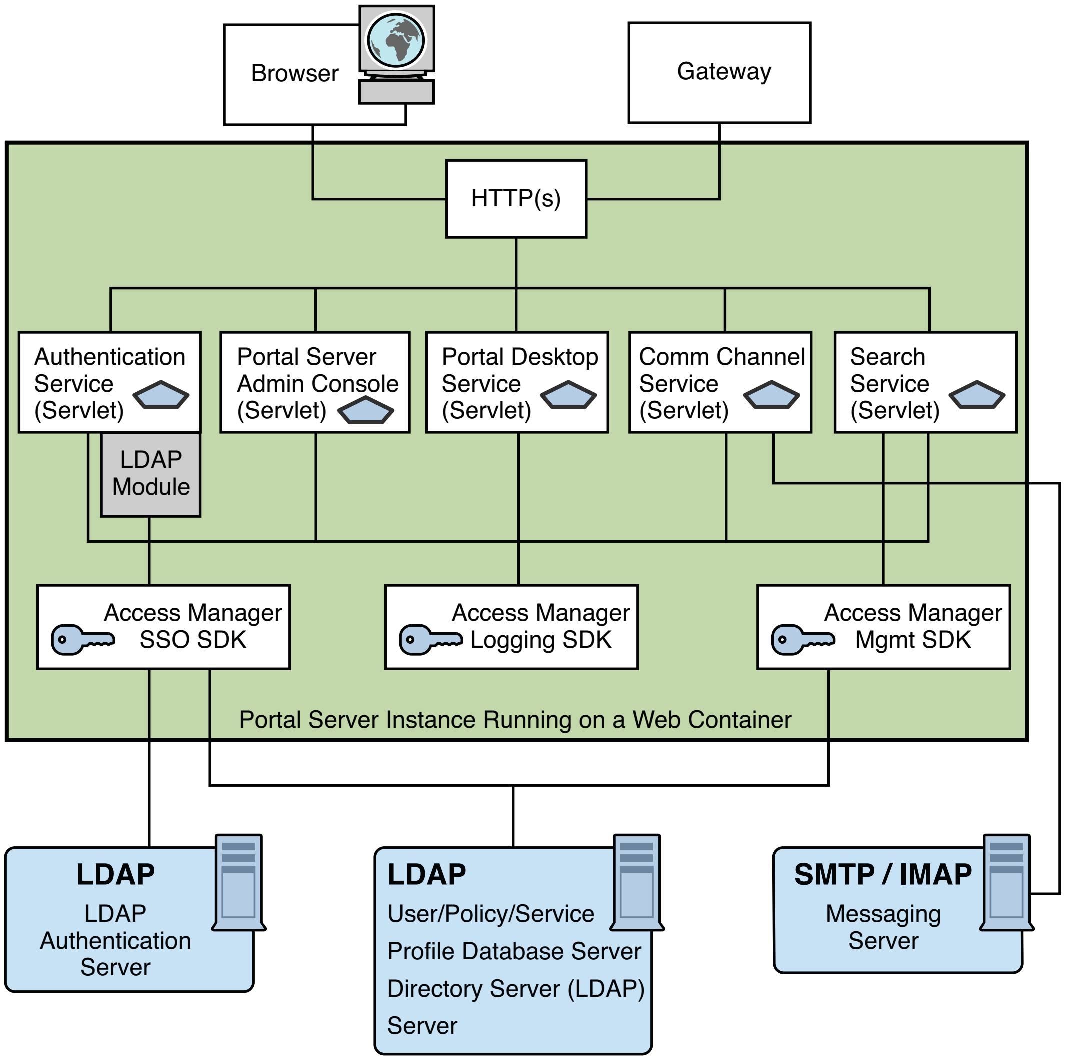

Browser
Gateway
HTTP(s)
Authentication Service (Servlet)
Portal Server Admin Console (Servlet)
Portal Desktop Service (Servlet)
Comm Channel Service (Servlet)
Search Service (Servlet)
LDAP Module
Access Manager SSO SDK
Access Manager Logging SDK
Access Manager Mgmt SDK
Portal Server Instance Running on a Web Container
LDAP
LDAP Authentication Server
LDAP
User/Policy/Service Profile Database Server Directory Server (LDAP) Server
SMTP / IMAP
Messaging Server

FIGURE 4–1 Portal Server Communication Links

In this figure, the box encloses the Portal Server instance running on Web Server technology. Within the instance are five servlets (Authentication, Portal Server administration console, Portal Desktop, Communication Channel, and Search), and the three SDKs (Access Manager SSO, Access Manager Logging, and Access Manager Management). The Authentication service servlet also makes use of an LDAP service provider module.

User traffic is directed to the appropriate servlet. Communication occurs between the Authentication service's LDAP module and the LDAP authentication server; between the

Communications channel servlet and the SMTP/IMAP messaging server; between the Access Manager SSO SDK and the LDAP server; and between the Access Manager Management SDK and the LDAP server.

Figure 4–1 shows that if the following processes or communication links fail, the portal solution becomes unavailable to end users:

- **Portal Server Instance**. Runs in the context of a web container. Components within an instance communicate through the JVM using Java APIs. An instance is a fully qualified domain name and a TCP port number. Portal Server services are web applications that are implemented as servlets or JSP™ files

  Portal Server is built on top of Access Manager for authentication single sign-on (session) management, policy, and profile database access. Thus, Portal Server inherits all the benefits (and constraints) of Access Manager with respect to availability and fault tolerance.

  By design, Access Manager's services are either stateless or the services can share context data. Services can recover to the previous state in case of a service failure.

  Within Portal Server, Portal Desktop does not share state data among instances. This means that an instance redirect causes the user context to be rebuilt for the enabled services. Usually, redirected users do not notice this because Portal Server services can rebuild a user context from the user's profile, and by using contextual data stored in the request. While this statement is generally true for out-of-the-box services, it might not be true for channels or custom code. Developers need to be careful to not design stateful channels to avoid loss of context upon instance failover.
- **Profile Database Server.** The profile database server is implemented by Directory Server software. Although this server is not strictly part of Portal Server, availability of the server and integrity of the database are fundamental to the availability of the system.
- **Authentication Server.** This is the directory server for LDAP authentication (usually, the same server as the profile database server). You can apply the same high availability techniques to this server as for the profile database server.
- **Secure Remote Access Gateway and Proxies.** The Secure Remote Access Gateway is a stand-alone Java technology process that can be considered stateless, because state information can be rebuilt transparently to end users. The Gateway profile maintains a list of Portal Server instances and does round robin load balancing across the Gateway instances. Session stickiness is not required in front of a Gateway, but with session stickiness, performance is better. On the other hand, session stickiness to Portal Server instances is enforced by Secure Remote Access.

  Secure Remote Access includes other Java technology processes called Netlet Proxy and Rewriter Proxy. You use these proxies to extend the security perimeter from behind the firewall, and limit the number of holes in the DMZ. You can install these proxies on separate nodes.

Example Portal Server Logical Architectures

This section provides some examples of logical architectures for Portal Server:

- "A Typical Portal Server Installation" on page 71
- "Portal Server Building Modules" on page 73
- "Access Manager and Portal Server on Separate Nodes" on page 80
- "Two Portal Servers One Access Manager" on page 81
- "Two Portal Servers Two Access Managers" on page 81
- "One Load Balancer Two Access Managers" on page 82

A Typical Portal Server Installation

Figure 4–2 illustrates some of the components of a portal deployment but does not address the actual physical network design, single points of failure, nor high availability.

This illustration shows the high-level architecture of a typical installation at a company site for a business-to-employee portal. In this figure, the Gateway is hosted in the company's demilitarized zone (DMZ) along with other systems accessible from the Internet, including proxy/cache servers, web servers, and mail Gateways. The portal node, portal search node, and directory server, are hosted on the internal network where users have access to systems and services ranging from individual employee desktop systems to legacy systems.

Note – If you are designing an ISP hosting deploymentthat hosts separate Portal Server instances for business customers who each want their own portal, contact your Sun representative. Portal Server requires customizations to provide ISP hosting functionality.

Figure 4–2 shows users on the Internet accessing the Gateway from a browser. The Gateway connects the user to the IP address and port for portal users attempting to access. For example, a B2B portal would usually allow access to only port 443, the HTTPS port. Depending on the authorized use, the Gateway forwards requests to the portal node, or directly to the service on the enterprise internal network.

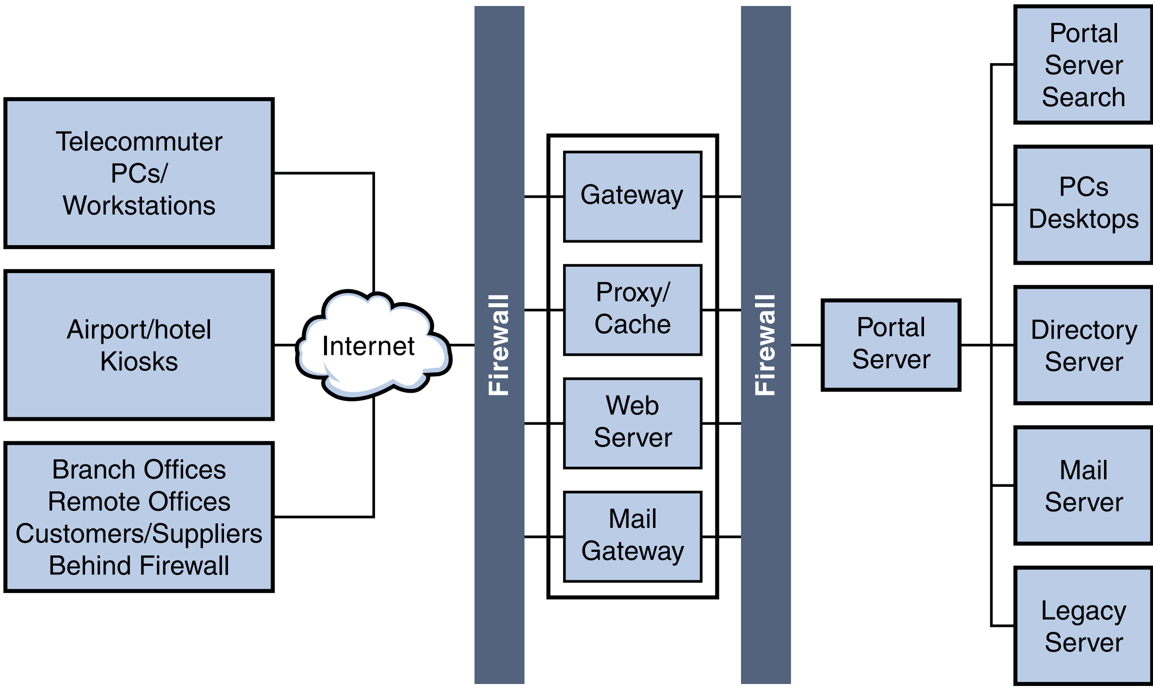


FIGURE 4–2 High-level Architecture for a Business-to-Employee Portal

Figure 4–3 shows a Portal Server deployment with Secure Remote Access services.

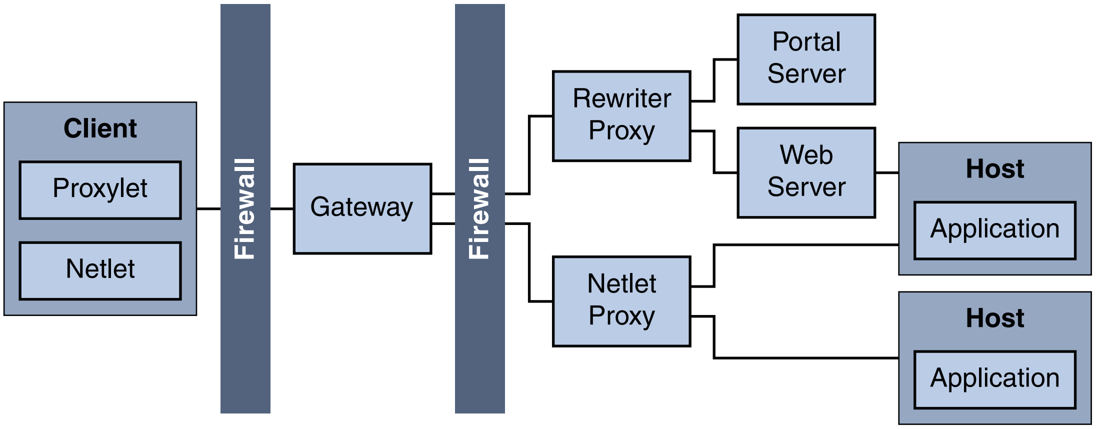


FIGURE 4–3 Secure Remote Access Deployment

Portal Server Building Modules

Because deploying Portal Server is a complex process involving many other systems, this section describes a specific configuration that provides optimum performance and horizontal scalability. This configuration is known as a Portal Server *building module*.

A Portal Server building module is a hardware and software construct with limited or no dependencies on shared services. A typical deployment uses multiple building modules to achieve optimum performance and horizontal scalability. Figure 4–4 shows the building module architecture.

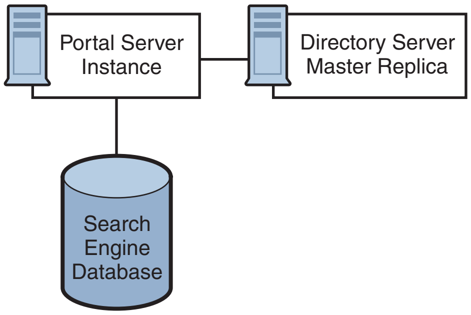


FIGURE 4–4 Portal Server Building Module Architecture

Note – The Portal Server building module is simply a recommended configuration. In some cases, a different configuration might result in slightly better throughput (usually at the cost of added complexity). For example, adding another instance of Portal Server to a four CPU system might result in up to ten percent additional throughput, at the cost of requiring a load balancer even when using just a single system.

Building Modules and High Availability Scenarios

Portal Server provides three scenarios for high availability:

- "Best Effort" on page 75

  The system is available as long as the hardware does not fail and as long as the Portal Server processes can be restarted by the watchdog process.
- "No Single Point of Failure" on page 75

  The use of hardware and software replication creates a deployment with no single point of failure (NSPOF). The system is always available, as long as no more than one failure occurs consecutively anywhere in the chain of components. However, in the case of failures, user sessions are lost.
- "Transparent Failover" on page 77

The system is always available but in addition to NSPOF, failover to a backup instance occurs transparently to end users. In most cases, users do not notice that they have been redirected to a different node or instance. Sessions are preserved across nodes so that users do not have to reauthenticate. Portal Server services are stateless or use checkpointing mechanisms to rebuild the current execution context up to a certain point. For detailed configuration information on Portal Server in high-availability and session failover, see the *Sun Java System Portal Server 7.1 Configuration Guide*.

Possible supported architectures include the following:

- Using Sun™ Cluster software on components that support Sun Cluster agents
- Multi-master Directory Server techniques

This section explains implementing these architectures and leverages the building module concept, from a high-availability standpoint.

Table 4–1 summarizes these high availability scenarios along with their supporting techniques.

TABLE 4–1 Portal Server High Availability Scenarios

Component Requirements	Necessary for Best Effort Deployment?	Necessary for NSPOF Deployment?	Necessary for Transparent Failover Deployment?
Hardware Redundancy	Yes	Yes	Yes
Portal Server Building Modules	No	Yes	Yes
Multi-master Configuration	No	Yes	Yes
Load Balancing	Yes	Yes	Yes
Stateless Applications and Checkpointing Mechanisms	No	No	Yes
Session Failover	No	No	Yes
Directory Server Clustering	No	No	Yes

Note – Load balancing is not provided out-of-the-box with the Sun Java System Web Server product.

Best Effort

In this scenario, you install Portal Server and Directory Server on a single node that has a secured hardware configuration for continuous availability, such as Sun Fire UltraSPARC™ III machines. (Securing a Solaris™ Operating Environment system requires that changes be made to its default configuration.)

This type of server features full hardware redundancy, including: redundant power supplies, fans, system controllers; dynamic reconfiguration; CPU hot-plug; online upgrades; and disks rack that can be configured in RAID 0+1 (striping plus mirroring), or RAID 5 using a volume management system, which prevents loss of data in case of a disk crash. Figure 4–5 shows a small, best effort deployment using the building module architecture.

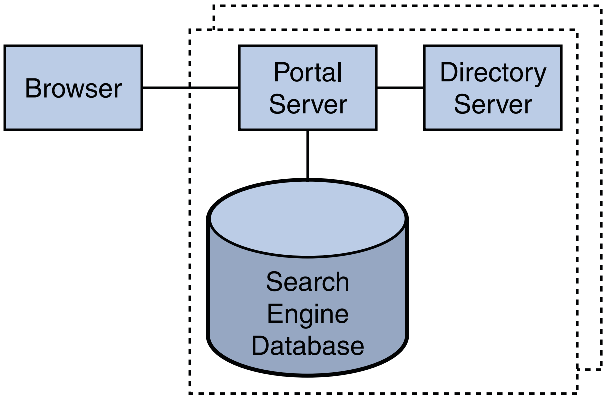


FIGURE 4–5 Best Effort Scenario

In this scenario, for memory allocation, four CPUs by eight GB RAM (4x8) of memory is sufficient for one building module. The Portal Server console is outside of the building module so that it can be shared with other resources. (Your actual sizing calculations might result in a different allocation amount.)

This scenario might suffice for task critical requirements. Its major weakness is that a maintenance action necessitating a system shutdown results in service interruption.

When Secure Remote Access is used, and a software crash occurs, a watchdog process automatically restarts the Gateway, Netlet Proxy, and Rewriter Proxy.

No Single Point of Failure

Portal Server natively supports the no single point of failure (NSPOF) scenario. NSPOF is built on top of the best effort scenario, and in addition, introduces replication and load balancing.

Figure 4–6 shows a building module consisting of a Portal Server instance, a Directory Server replica for profile reads and a search engine database. As such, at least two building modules are necessary to achieve NSPOF, thereby providing a backup if one of the building modules fails. These building modules consist of four CPUs by eight GB RAM.

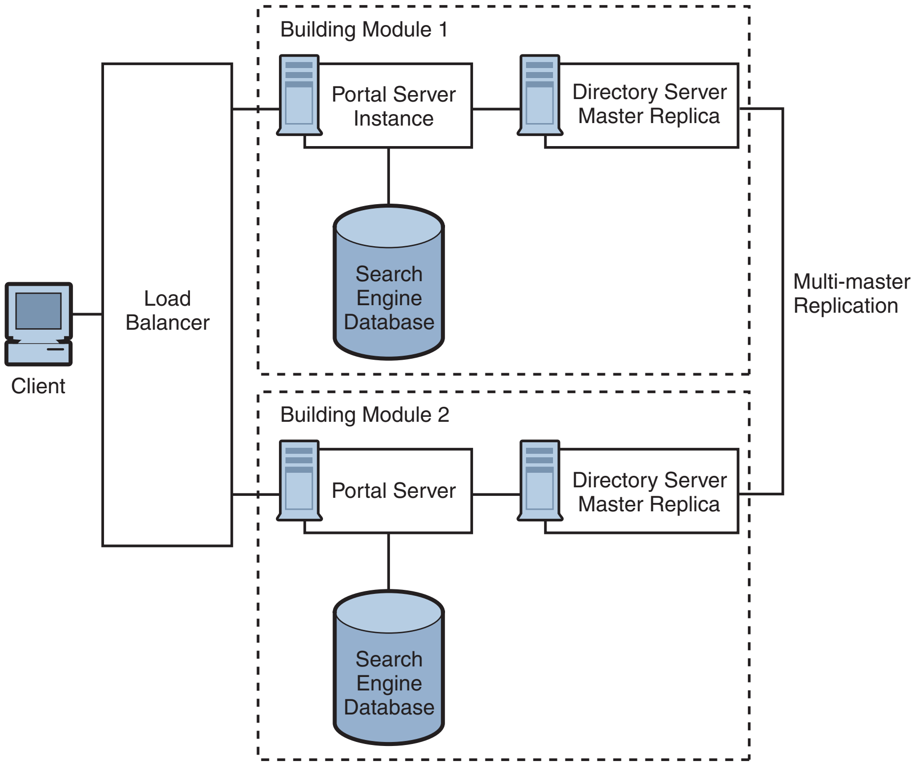


FIGURE 4–6 No Single Point of Failure Example

When the load balancer detects Portal Server failures, it redirects users' requests to a backup building module. Accuracy of failure detection varies among load balancing products. Some products are capable of checking the availability of a system by probing a service involving several functional areas of the server, such as the servlet engine, and the JVM. In particular, most vendor solutions from Resonate, Cisco, Alteon, and others enable you to create arbitrary scripts for server availability. As the load balancer is not part of the Portal Server software, you must acquire it separately from a third-party vendor.

Note – Access Manager requires that you set up load balancing to enforce sticky sessions. This means that once a session is created on a particular instance, the load balancer needs to always return to the same instance for that session. The load balancer achieves this by binding the session cookie with the instance name identification. In principle, that binding is reestablished when a failed instance is decommissioned. Sticky sessions are also recommended for performance reasons.

Multi-master replication (MMR) takes places between the building modules. The changes that occur on each directory are replicated to the other, which means that each directory plays both roles of supplier and consumer. For more information on MMR, refer to the *Sun Java System Directory Server Deployment Guide*.

Note – In general, the Directory Server instance in each building module is configured as a replica of a master directory, which runs elsewhere. However, nothing prevents you from using a master directory as part of the building module. The use of masters on dedicated nodes does not improve the availability of the solution. Use dedicated masters for performance reasons.

Redundancy is equally important to the directory master so that profile changes through the administration console or the Portal Desktop, along with consumer replication across building modules, can always be maintained. Portal Server and Access Manager support MMR. The NSPOF scenario uses a multi-master configuration. In this configuration, two suppliers can accept updates, synchronize with each other, and update all consumers. The consumers can refer update requests to both masters.

Secure Remote Access follows the same replication and load balancing pattern as Portal Server to achieve NSPOF. As such, two Secure Remote Access Gateways and pair of proxies are necessary in this scenario. The Secure Remote Access Gateway detects a Portal Server instance failure when the instance does not respond to a request after a certain time-out value. When this occurs, the HTTPS request is routed to a backup server. The Secure Remote Access Gateway performs a periodic check for availability until the first Portal Server instance is up again.

The NSPOF high availability scenario is suitable to business critical deployments. However, some high availability limitations in this scenario might not fulfill the requirements of a mission critical deployment.

Transparent Failover

Transparent failover uses the same replication model as the NSPOF scenario but provides additional high availability features, which make the failover to a backup server transparent to end users.

Figure 4–7 shows a transparent failover scenario. Two building modules are shown, consisting of four CPUs by eight GB RAM. Load balancing is responsible for detecting Portal Server failures and redirecting users' requests to a backup Portal Server in the building module. Building Module 1 stores sessions in the sessions repository. If a crash occurs, the application server retrieves sessions created by Building Module 1 from the sessions repository.

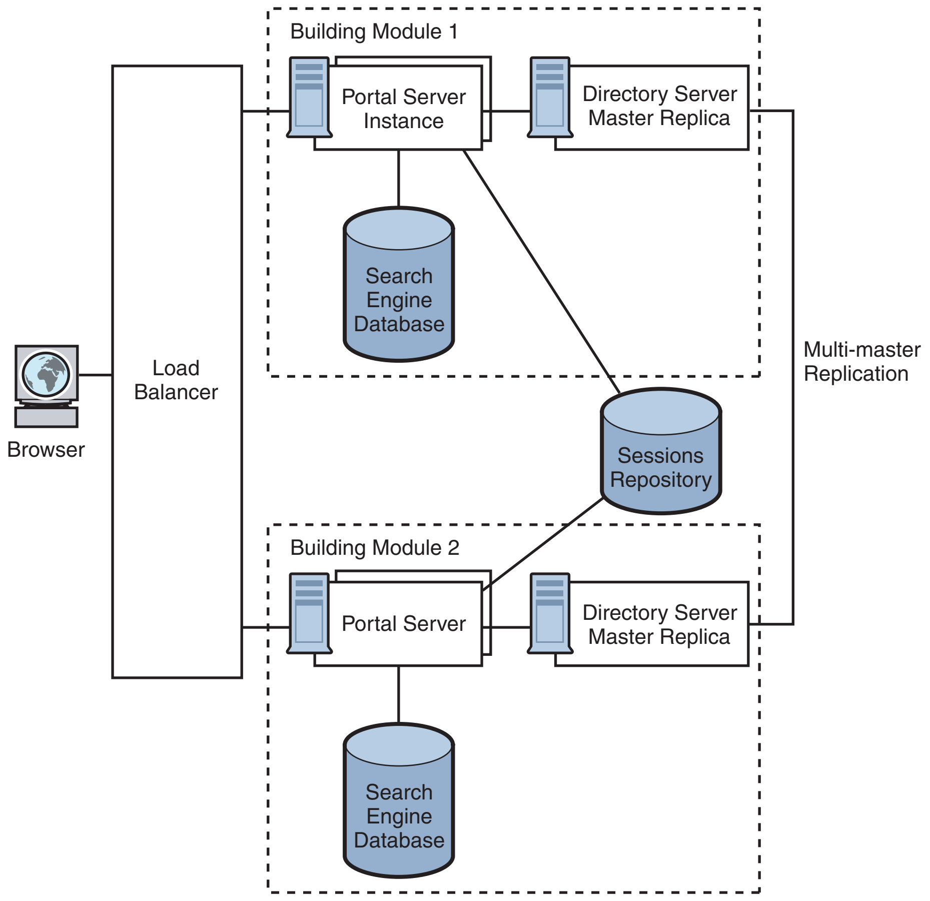


FIGURE 4–7 Transparent Failover Example Scenario

The session repository is provided by the application server software. Portal Server is running in an application server. Portal Server supports transparent failover on application servers that support HttpSession failover. See Appendix A for more information.

With session failover, users do not need to reauthenticate after a crash. In addition, portal applications can rely on session persistence to store context data used by the checkpointing. You configure session failover in the `AMConfig.properties` file by setting the `com.iplanet.am.session.failover.enabled property` to **`true`**.

The Netlet Proxy cannot support the transparent failover scenario because of the limitation of the TCP protocol. The Netlet Proxy tunnels TCP connections, and you cannot migrate an open TCP connection to another server. A Netlet Proxy crash drops off all outstanding connections that would have to be reestablished.

Building Module Solution Recommendations

This section describes guidelines for deploying your building module solution.

How you construct your building module affects performance.

Consider the following recommendations to deploy your building module properly:

- Deploy a building module on a single machine.
- If you use multiple machines, or if your Portal Server machine is running a large number of instances, use a fast network interconnect.
- On servers with more than eight CPUs, create processor sets or domains with either two or four CPUs. For example, if you choose to install two instances of Portal Server on an eight CPU server, create two four-CPU processor sets.

Directory Server

Identify your Directory Server requirements for your building module deployment. For specific information on Directory Server deployment, see the *Directory Server Deployment Guide*.

Consider the following Directory Server guidelines when you plan your Portal Server deployment:

- The amount of needed CPU in the Directory Server consumer replica processor set depends on the number of Portal Server instances in the building module as well as performance and capacity considerations.
- If possible, dedicate a Directory Server instance for the sole use of the Portal Server instances in a building module.
- Map the entire directory database indexes and cache in memory to avoid disk latency issues.
- When deploying multiple building modules, use a multi-master configuration to work around bottlenecks caused by the profile updates and replication overhead to the Directory Server supplier.

LDAP

The scalability of building modules is based on the number of LDAP writes resulting from profile updates and the maximum size of the LDAP database.

Note – If the LDAP server crashes with the `_db` files in the `/tmp` directory, the files are lost when the server restarts. This improves performance but also affects availability.

If the analysis at your specific site indicates that the number of LDAP write operations is indeed a constraint, some of the possible solutions include creating building modules that replicate only a specific branch of the directory and a layer in front that directs incoming requests to the appropriate instance of portal.

Search Service

When you deploy the Search Engine as part of your building module solution, consider the following:

- In each building module, make sure only one Portal Server instance has the Search Engine database containing the Resource Descriptors (RDs). The remaining Portal Server instances have default empty Search Engine databases.
- Factors that influence whether to use a building module for the portal Search database include the intensity of search activities in a Portal Server deployment, the range of search hits, and the average number of search hits for all users, in addition to the number of concurrent searches. For example, the load generated on a server by the Search Engine can be both memory and CPU intensive for a large index and heavy query load.
- You can install Search on a machine separate from Portal Server, to keep the main server dedicated to portal activity. When you do so, you use the `searchURL` property of the Search provider to point to the second machine where Search is installed. The Search instance is a normal portal instance. You install the Search instance just as you do the portal instance, but use it just for Search functionality.

Access Manager and Portal Server on Separate Nodes

Figure 4–8 illustrates Access Manager and Portal Server residing on separate nodes.

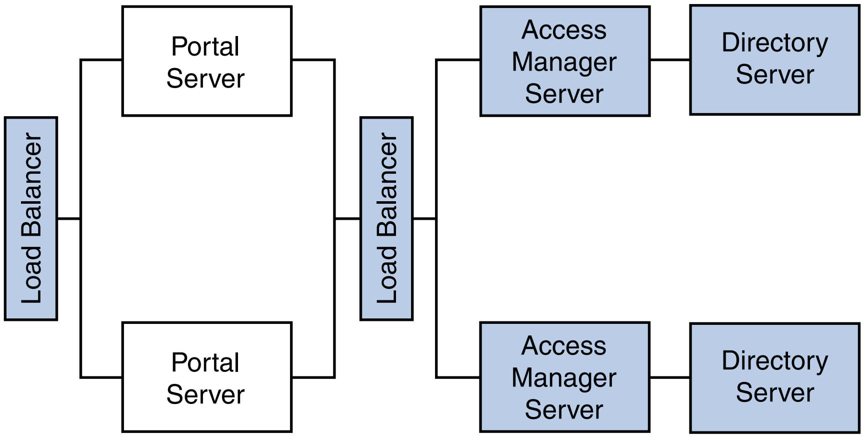


FIGURE 4–8 Access Manager and Portal Server on Different Nodes

As a result of this implementation of Portal Server and Access Manager separation, other topology permutations are possible for portal services architecture deployments as shown in the next three figures.

Two Portal Servers One Access Manager

Figure 4–9 shows two Portal Server instances configured to work with a single Access Manager and two Directory Servers where both the Access Manager and the Directory Servers operate in a Java Enterprise System Sun Clustered environment. This configuration is ideal when Access Manager and Directory Server instances are not the bottleneck.

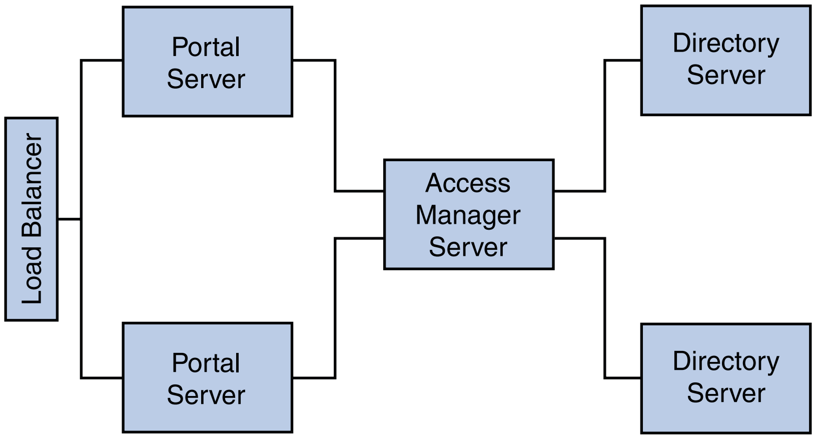


FIGURE 4–9 Two Portal Servers and One Access Manager

Two Portal Servers Two Access Managers

Figure 4–10 shows a configuration for maximum horizontal scalability and higher availability achieved by a horizontal server farm. Two Portals Servers can be fronted with a load balancer for maximum throughput and high availability.

Another load balancer can be put between Portal Servers and Access Managers to achieve authentication and policy processes as a load distributor and failover mechanism for higher availability.

In this scenario, Blade 1500s can be utilized for Portal Services to distribute the load, similar Blades can be used to host Access Manager Services and Directory Services respectively. With the architecture shown in Figure 4–10, a redundancy of services exists for each of the product stack, therefore, most of the unplanned downtime can be minimized or eliminated.

However, the planned downtime is still an issue. If an upgrade or patch includes changes to the Directory Server software schema used by the Access Manager software, all of the software components must be stopped to update the schema information stored in the Directory Server. However, updating schema information can be considered a fairly rare occurrence in most patch upgrades.

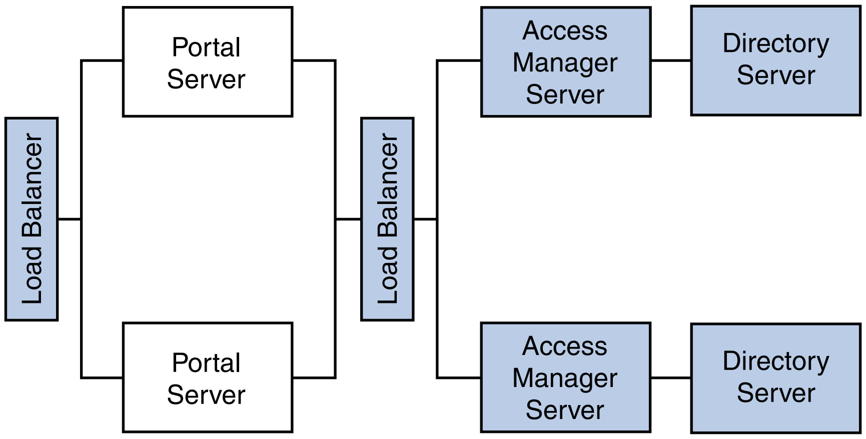


FIGURE 4–10 Two Portal Servers and Two Access Managers

One Load Balancer Two Access Managers

Figure 4–11 shows configuration allowing authentication throughput coming from Portal Server to be load-balanced across the two Access Managers.

This configuration could be implemented when the Portal Server resides on a high-end medium to large server (that is 1 to 4 processors) with a very wide bandwidth network connection. The Access Managers with the policy and authentication services could be on two medium-size servers.

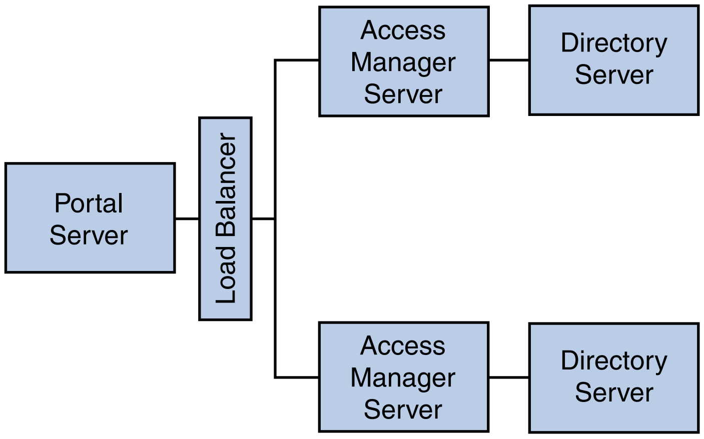


FIGURE 4–11 Load Balancing two Access Managers

Example Secure Remote Access Logical Architectures

The Secure Remote Access Gateway provides the interface and security barrier between the remote user sessions originating from the Internet and your organization's intranet. The Gateway serves two main functions:

- Provides basic authentication services to incoming user sessions, including establishing identity and allowing or denying access to the platform.
- Provides mapping and rewriting services to enable web-based links to the intranet content for users.

For Internet access, use 128-bit SSL to provide the best security arrangement and encryption or communication between the user's browser and Portal Server. The Gateway, Netlet, NetFile, Netlet Proxy, Rewriter Proxy, and Proxylet constitute the major components of Secure Remote Access.

This section lists some of the possible configurations of these components. This section is meant only as a guide, not a complete deployment reference. Choose a configuration based on your business needs:

- "Basic Secure Remote Access Configuration" on page 83
- "Disable Netlet" on page 84
- "Proxylet" on page 85
- "Multiple Gateway Instances" on page 86
- "Netlet and Rewriter Proxies" on page 87
- "Netlet and Rewriter Proxies on Separate Nodes" on page 89
- "Two Gateways and Netlet Proxy" on page 89
- "Gateway with Accelerator" on page 90
- "Netlet with Third-Party Proxy" on page 91
- "Reverse Proxy" on page 92

Tip – To set up the authlessanonymous page to display through the Gateway, add `/portal/dt` to the non-authenticated URLs of the gateway profile. However, this means that even for normal users, portal pages will not need authentication and no session validation is performed.

Basic Secure Remote Access Configuration

Figure 4–12 shows the most simple configuration possible for Secure Remote Access. The figure shows a client browser running NetFile and Netlet. The Gateway is installed on a separate machine in the DMZ between two firewalls. The Portal Server is located on a machine beyond the second firewall in the intranet. The other application hosts that the client accesses are also located beyond the second firewall in the intranet.

The Gateway is in the DMZ with the external port open in the firewall through which the client browser communicates with the Gateway. In the second firewall, for HTTP or HTTPS traffic, the Gateway can communicate directly with internal hosts. If security policies do not permit it, use Secure Remote Access proxies between the Gateway and the internal hosts. For Netlet traffic, the connection is direct from the Gateway to the destination host.

Without a Secure Remote Access proxy, the SSL traffic is limited to the Gateway and the traffic is unencrypted from the Gateway to the internal host (unless the internal host is running in HTTPS mode). Any internal host to which the Gateway has to initiate a Netlet connection should be directly accessible from DMZ. This can be a potential security problem and hence this configuration is recommended only for the simplest of installations.

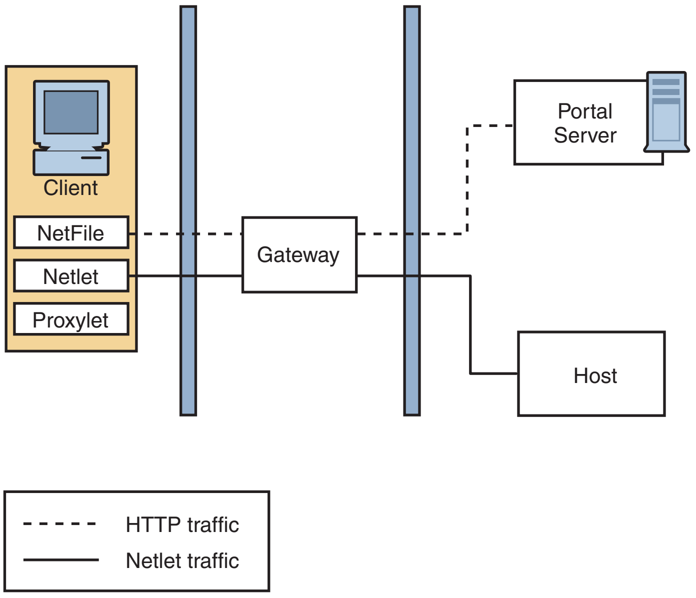


FIGURE 4–12 Basic Secure Remote Access Configuration

Disable Netlet

Figure 4–13 shows a scenario similar to the basic Secure Remote Access configuration except that Netlet is disabled. If the client deployment is not going to use Netlet for securely running applications that need to communicate with intranet, then use this setup for performance improvement.

You can extend this configuration and combine it with other deployment scenarios to provide better performance and a scalable solution.

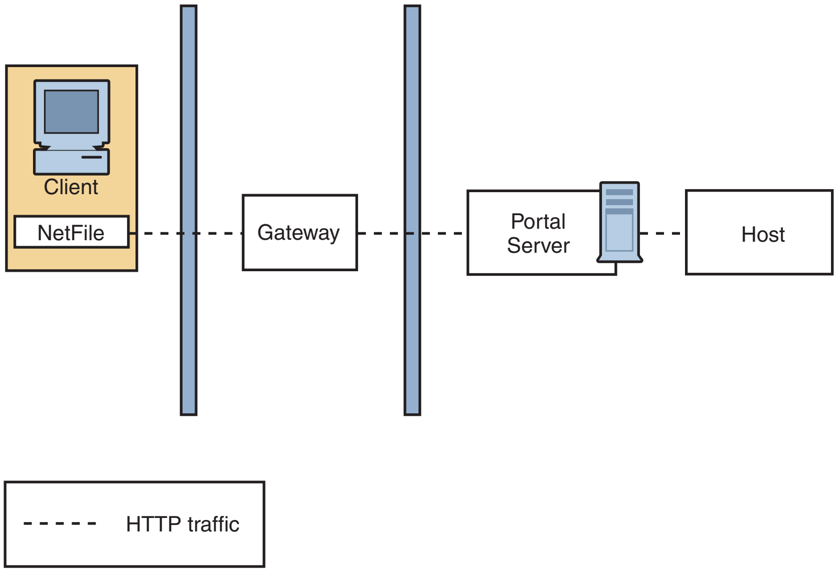


FIGURE 4–13 Disable Netlet

Proxylet

Figure 4–14 illustrates how Proxylet enables users to securely access intranet resources through the Internet without exposing these resources to the client.

It inherits the transport mode (either HTTP or HTTPS) from the Gateway.

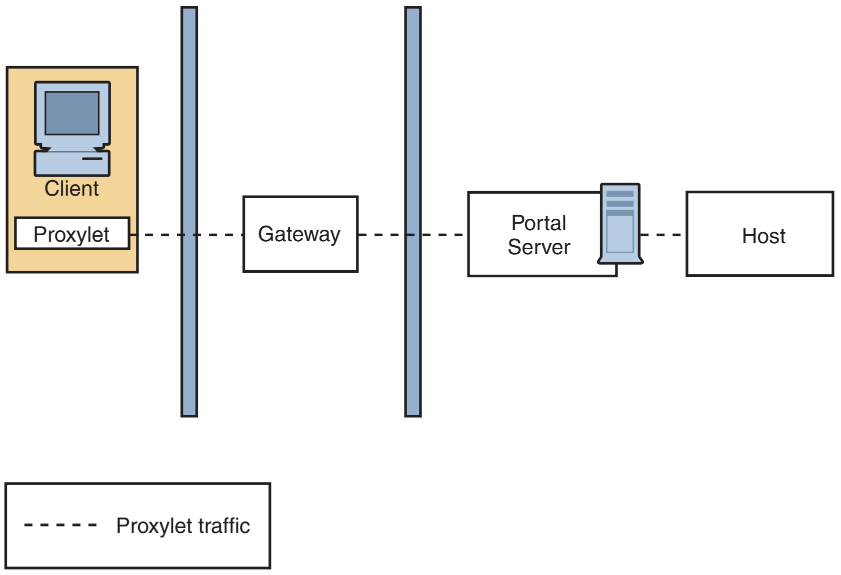


FIGURE 4–14 Proxylet

Multiple Gateway Instances

Figure 4–15 shows an extension of the Secure Remote Access basic configuration. Multiple Gateway instances run on the same machine or multiple machines. You can start multiple Gateway instances with different profiles.

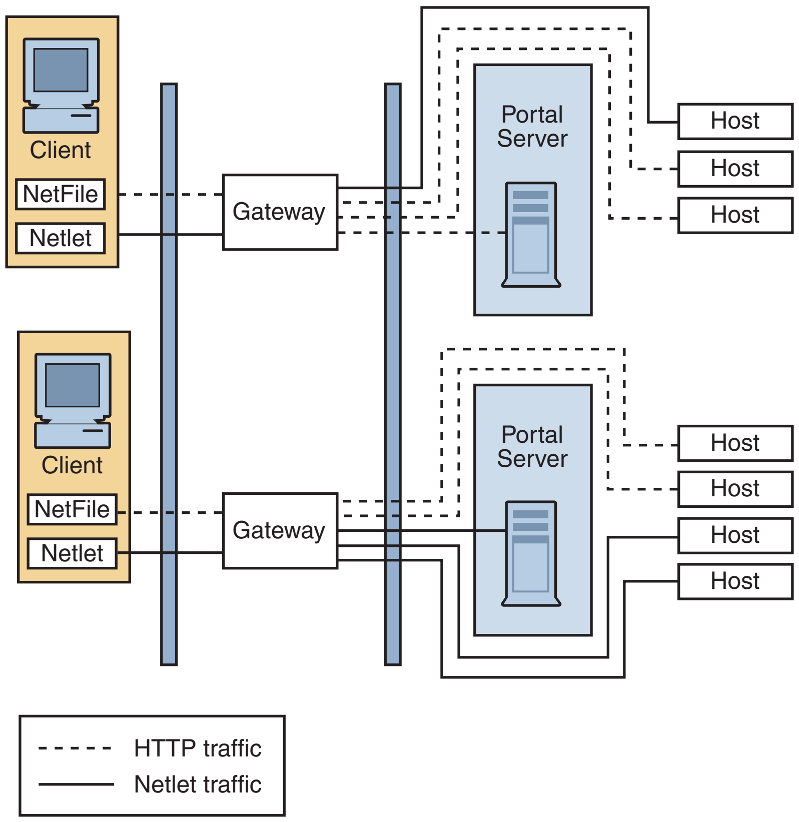


FIGURE 4–15 Multiple Gateway Instances

Note – Although Figure 4–15 shows a 1-to-1 correspondence between the Gateway and the Portal Servers, this need not necessarily be the case in a real deployment. You can have multiple Gateway instances, and multiple Portal Server instances, and any Gateway can contact any Portal Server depending on the configuration.

The disadvantage to this configuration is that multiple ports need to be opened in the second firewall for each connection request. This could cause potential security problems.

Netlet and Rewriter Proxies

Figure 4–16 shows a configuration with a Netlet Proxy and a Rewriter Proxy. With these proxies, only two open ports are necessary in the second firewall.

The Gateway need not contact the application hosts directly now, but will forward all Netlet traffic to the Netlet proxy and Rewriter traffic to the Rewriter Proxy. Since the Netlet Proxy is within the intranet, it can directly contact all the required application hosts without opening multiple ports in the second firewall.

The traffic between the Gateway in the DMZ and the Netlet Proxy is encrypted, and gets decrypted only at the Netlet Proxy, thereby enhancing security.

If the Rewriter Proxy is enabled, all traffic is directed through the Rewriter Proxy, irrespective of whether the request is for the Portal Server node or not. This ensures that the traffic from the Gateway in the DMZ to the intranet is always encrypted.

Because the Netlet Proxy, Rewriter Proxy, and Portal Server are all running on the same node, there might be performance issues in such a deployment scenario. This problem is overcome when proxies are installed on a separate nodes to reduce the load on the Portal Server node.

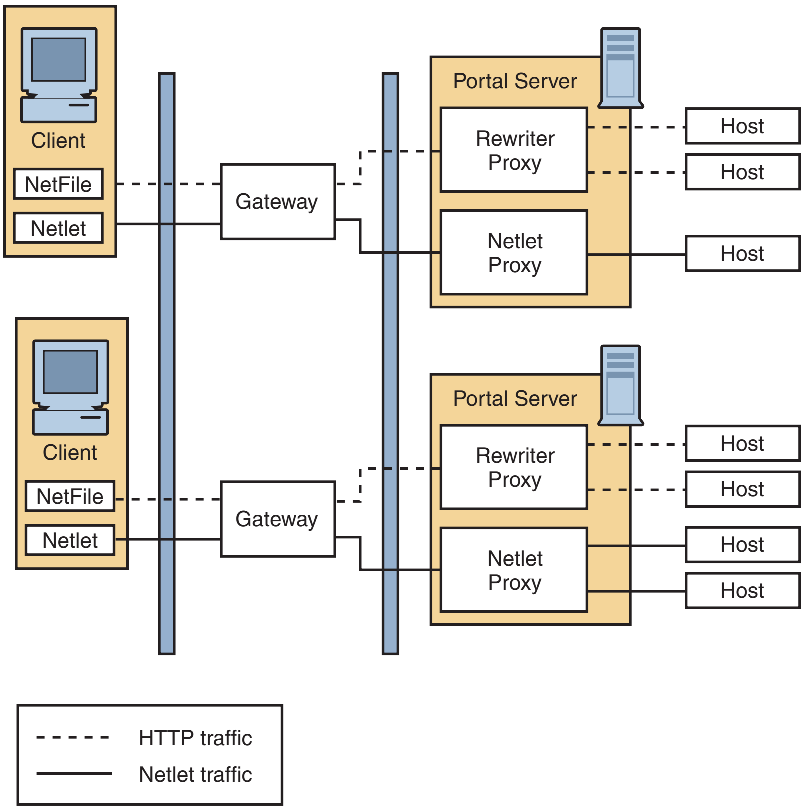


FIGURE 4–16 Netlet and Rewriter Proxies

Netlet and Rewriter Proxies on Separate Nodes

To reduce the load on the Portal Server node and still provide the same level of security at increased performance, you can install Netlet and Rewriter Proxies on separate nodes. This deployment has an added advantage in that you can use a proxy and shield the Portal Server from the DMZ. The node that runs these proxies needs to be directly accessible from the DMZ.

Figure 4–17 shows the Netlet Proxy and Rewriter Proxy on separate nodes. Traffic from the Gateway is directed to the separate node, which in turn directs the traffic through the proxies and to the required intranet hosts.

You can have multiple instances or installations of Netlet and Rewriter Proxies. You can configure each Gateway to try to contact various instances of the proxies in a round robin manner depending on availability.

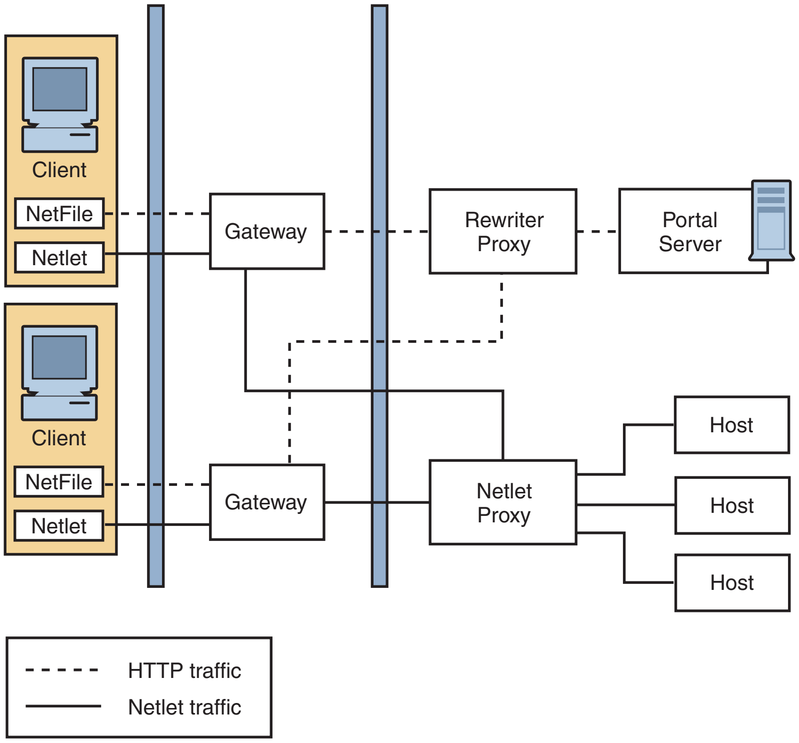


FIGURE 4–17 Proxies on Separate Nodes

Two Gateways and Netlet Proxy

Load balancers provide a failover mechanism for higher availability for redundancy of services on the Portal Servers and Access Managers.

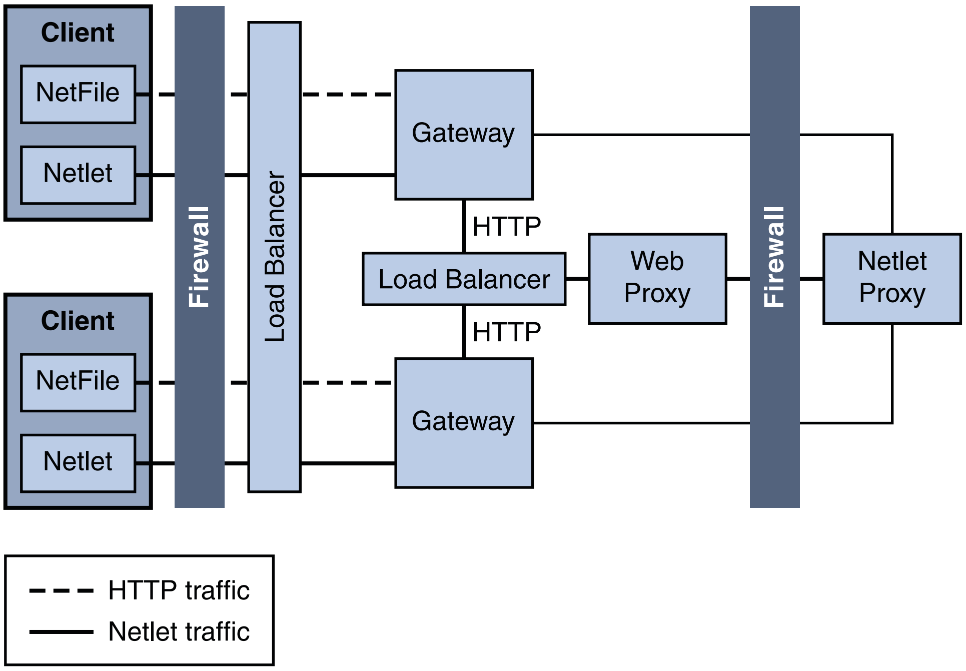


FIGURE 4–18 Two Gateways and Netlet Proxy

Gateway with Accelerator

You can configure an external SSL device to run in front of the Gateway in open mode. It provides the SSL link between the client and Secure Remote Access. For information on accelerators, see the *Sun Java System Portal Server 7.1 Secure Remote Access Administration Guide*.

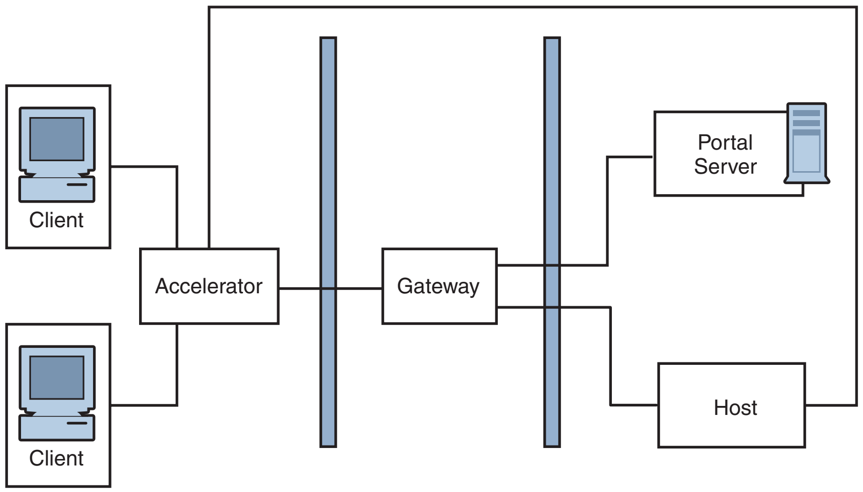


FIGURE 4–19 Secure Remote Access Gateway with External Accelerator

Netlet with Third-Party Proxy

Figure 4–20 illustrates using a third-party proxy to limit the number of ports in the second firewall to one. You can configure the Gateway to use a third-party proxy to reach the Rewriter and the Netlet Proxies.

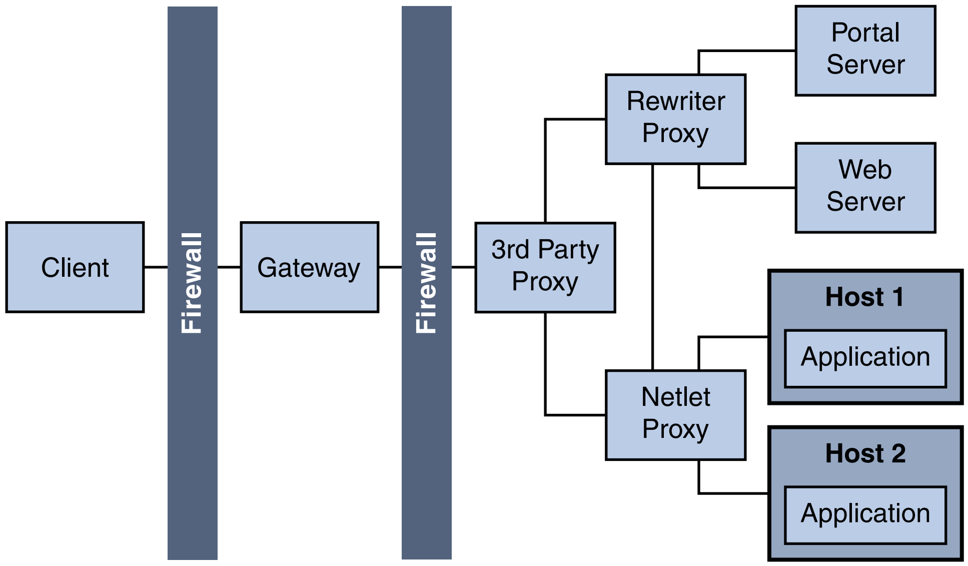


FIGURE 4–20 Netlet and Third-Party Proxy

Reverse Proxy

A proxy server serves Internet content to the intranet, while a reverse proxy serves intranet content to the Internet. Certain deployments of reverse proxy are configured to serve the Internet content to achieve load balancing and caching.

Figure 4–21 illustrates how you can configure a reverse proxy in front of the Gateway to serve both Internet and intranet content to authorized users. Whenever the Gateway serves web content, it needs to ensure that all subsequent browser requests based on this content are routed through the Gateway. This is achieved by identifying all URLs in this content and rewriting as appropriate.

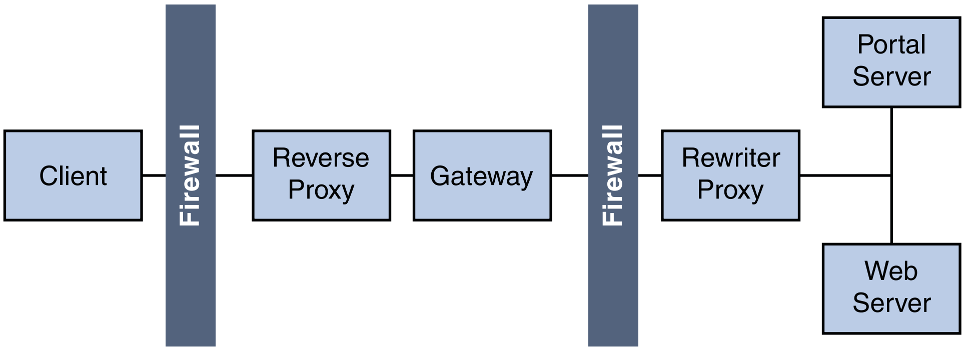


FIGURE 4–21 Using a Reverse Proxy in Front of the Gateway

Deployment Scenario

The completed logical architecture design by itself is not sufficient to move forward to the deployment design phase of the solution life cycle. You need to pair the logical architecture with the quality of service (QoS) requirements determined during the technical requirements phase. The pairing of the logical architecture with the QoS requirements constitutes a deployment scenario.

The deployment scenario is the starting point for designing the deployment architecture, as explained in Chapter 5.

◆ ◆ ◆ CHAPTER 5

Planning Your Deployment Design

During the deployment design phase of the solution life cycle, you design a high-level deployment architecture and a low-level implementation specification, and prepare a series of plans and specifications necessary to implement the solution. Project approval occurs in the deployment design phase.

This chapter contains the following sections:

- "About Deployment Design" on page 93
- "Estimating Processor Requirements" on page 94
- "Identifying Performance Bottlenecks" on page 97
- "Optimizing Resources" on page 98
- "Managing Risks" on page 99

- "About Deployment Design" on page 93
- "Estimating Processor Requirements" on page 94
- "Identifying Performance Bottlenecks" on page 97
- "Optimizing Resources" on page 98
- "Managing Risks" on page 99

About Deployment Design

Deployment design begins with the deployment scenario created during the logical design and technical requirements phases of the solution life cycle. The deployment scenario contains a logical architecture and the quality of service requirements for the solution. You map the components identified in the logical architecture across physical servers and other network devices to create a deployment architecture. The quality of service requirements provide guidance on hardware configurations for performance, availability, scalability, and other related quality of service specifications.

Designing the deployment architecture is an iterative process. You typically revisit the quality of service requirements and reexamine your preliminary designs. You take into account the interrelationship of the quality of service requirements, balancing the trade-offs and cost of ownership issues to arrive at an optimal solution that ultimately satisfies the business goals of the project.

The best way to start deployment planning is to begin with a reference or building block architecture which is already well documented and tested. It is easier to modify this into what's required rather than starting from scratch. Factors that contribute to successful deployment design are past design experience, knowledge of systems architecture, domain knowledge, and applied creative thinking.

Deployment design typically revolves around achieving performance requirements while meeting other quality of service requirements. The strategies you use must balance the trade-offs of your design decisions to optimize the solution. The methodology you use typically involves the following tasks:

1. **Estimate processor requirements**. Deployment design often begins with portal sizing in the logical architecture. Start with the use cases and modify your estimates accordingly. Also consider any previous experience you have with designing enterprise systems.
2. **Estimate processor requirements for secure transport**. Study use cases that require secure transport and modify CPU estimates accordingly.
3. **Replicate services for availability and scalability**. Make modifications to the design to account for quality of service requirements for availability and scalability. Consider load balancing solutions that address availability and failover considerations.
4. During your analysis, consider the trade-offs of your design decisions. For example, what affect does the availability and scalability strategy have on serviceability (maintenance) of the system? What are the others costs of the strategies?
5. **Identify bottlenecks.** Examine the deployment design to identify any bottlenecks that cause the transmission of data to fall beneath requirements, and make adjustments.
6. **Optimize resources**. Review your deployment design for resource management and consider options that minimizes costs while fulfilling requirements.
7. **Manage risks**. Revisit your business and technical analysis, and modify your design to account for events or situations that your earlier planning did not foresee.

Estimating Processor Requirements

With a baseline figure established in the usage analysis, you can then validate and refine that figure to account for scalability, high availability, reliability, and good performance:

▼ Steps to Estimate Processor Requirements

1. “Customize the Baseline Sizing Figures” on page 95
2. “Validate Baseline Sizing Figures” on page 96
3. “Refine Baseline Sizing Figures” on page 96
4. “Validate Your Final Figures” on page 97

   The following sections describe these steps.

Customize the Baseline Sizing Figures

Establishing an appropriate sizing estimate for your Portal Server deployment is an iterative process. You might wish to change the inputs to generate a range of sizing results. Customizing your Portal Server deployment can greatly affect its performance.

After you have an estimate of your sizing, consider:

- “LDAP Transaction Numbers” on page 95
- “Web Container Requirements” on page 95

LDAP Transaction Numbers

Use the following LDAP transaction numbers for an out-of-the-box portal deployment to understand the impact of the service demand on the LDAP master and replicas. These numbers change once you begin customizing the system.These numbers are taken for the Developer Sample and include access manager work.

- Access to authless anonymous portal - 6 BINDS, 49 SRCH
- Login by using the Login channel - 1 BIND, 2 SRCH
- Removing a channel from the Portal Desktop - 14 SRCH, 2MOD
- Reloading the Portal Desktop - 0 ops

Web Container Requirements

One of the primary uses of Portal Server installed on an web container is to integrate portal providers J2EE technology stack constructs, such as Enterprise JavaBeans^{TM} in the case of the application server or things like Java Database Connectivity (JDBC) and LDAP (LDAPSDK) connectivity. These other applications and modules can consume resources and affect your portal sizing. It is best to use connection pooling and jndi tooling available through the web container.

Validate Baseline Sizing Figures

Now that you have an estimate of the number of CPUs for your portal deployment, use a trial deployment to measure the performance of the portal. Use load balancing and stress tests to determine:

- Throughput, the amount of data processed in a specified amount of time
- Latency, the period of time that one component is waiting for another component
- Maximum number of concurrent sessions

Portal samples are provided with the Portal Server. You can use them, with channels similar to the ones you will use, to create a load on the system. The samples are located on the Portal Desktop.

Use a trial deployment to determine your final sizing estimates. A trial deployment helps you to size back-end integration to avoid potential bottlenecks with Portal Server operations.

Refine Baseline Sizing Figures

Your next step is to refine your sizing figure. In this section, you build in the appropriate amount of headroom so that you can deploy a portal site that features scalability, high availability, reliability and good performance.

Because your baseline sizing figure is based on so many estimates, do not use this figure without refining it.

When you refine your baseline sizing figure:

- Use your baseline sizing figure as a reference point.
- Expect variations from your baseline sizing figure.
- Learn from the experience of others.
- Use your own judgement and knowledge.
- Examine other factors in your deployment.

  If the Portal Server deployment involves multiple data centers on several continents and even traffic, you need a higher final sizing figure than if you have two single data centers on one continent with heavy traffic.
- Plan for changes.

  A portal site is likely to experience various changes after you launch it. Changes you might encounter include the following:
  - An increase in the number of channels
  - Growth in the user base
  - Modification of the portal site's purpose

- Changes in security needs
- Power failures
- Maintenance demands

  Considering these factors enables you to develop a sizing figure that is flexible and enables you to avoid risk when your assumptions regarding your portal change following deployment.

  The resulting figure ensures that your portal site has:

- Scalability high availability, reliability and high performance
- Room for whatever you want to provide
- Flexibility for adjusting to changes

Validate Your Final Figures

Use a trial deployment to verify that the portal deployment satisfies your business and technical requirements.

Identifying Performance Bottlenecks

By identifying potential bottlenecks during the design phase, you can plan for your portal performance needs.

Before reading the section on memory consumption and its affect on performance, read the following document on tuning the Java Virtual Machine, version 1.4.2. If you are deploying on Sun Java System Application Server or Sun Java System Web Server, the Java Virtual machine will be version 1.5. However if you are deploying on a third party web container it could be version 1.4.2.

`http://java.sun.com/products/hotspot/index.html`

Memory Consumption and Garbage Collection

Portal Server requires substantial amounts of memory to provide the highest possible throughput. At initialization, a maximum address space is virtually reserved but does not allocate physical memory unless needed. The complete address space reserved for object memory (heap) can be divided into young generation (eden) and old space (tenured). As the name suggests young means a area reserved for newly created Java objects, and old refers to space allocated to Java objects that have been around for a while.

Most applications suggest using a larger percentage of the total heap for the new generation, but in the case of Portal Server, using only one eighth the space for the young generation is appropriate, because most memory used by Portal Server is long-lived. The sooner the memory is copied to the old generation the better the garbage collection (GC) performance.

By default, even with a large heap size, after a portal instance has been running under moderate load for a few days, most of the heap appears to be used because of the lazy nature of the GC. The GC performs full garbage collections until the resident set size (RSS) reaches approximately 85 percent of the total heap space; at that point the garbage collections can have a measurable impact on performance.

For example, on a 900 MHz UltraSPARCIII™, a full GC on a 2 GB heap can take over ten seconds. During that period of time, the system is unavailable to respond to web requests. During a reliability test, full GCs are clearly visible as spikes in the response time. In production, full GCs can be unnoticed but monitoring scripts that measure system performance need to account for the possibility that a full GC can occur frequently.

Measuring the frequency of full GCs as a indicator may sometimes be useful in determining if a memory leak is present in the web container. Conduct an analysis that shows the expected frequency (of a baseline system) and compare that to the observed rate of full GCs. To record the frequency of GCs, use the `vebose:gc` JVM™ parameter.

Optimizing Resources

You can optimize resources by using the following:

- “SSL Off-loading onto Load Balancer” on page 98
- “Sun Enterprise Midframe Line” on page 98

SSL Off-loading onto Load Balancer

SSL-intensive servers, such as the Secure Remote Access Gateway, require large amounts of processing power to perform the encryption required for each secure transaction. Using a load balancer with ssl capability can speed up the Portal Gateway's by off-loading the execution of cryptographic algorithms.

Terminating SSL traffic on a load balancer in a DMZ simplifies the portal topology. Access manager sessions are maintained in cookies and it is very important from a performance point of view that the correct servers are engaged in processing a browser request. For Example, those servers which have that particular session in its cache. Session stickiness is much easily achieved using cookies and http than using https to all the back-end servers.

Sun Enterprise Midframe Line

Normally, for a production environment, you would deploy Portal Server and Secure Remote Access on separate machines. However, in the case of the Sun Enterprise™ midframe machines, which support multiple hardware domains, you can install both Portal Server and Secure

Remote Access in different domains on the same Sun Enterprise midframe machine. The normal CPU and memory requirements that pertain to Portal Server and Secure Remote Access still apply; you would implement the requirements for each in the separate domains.

In this type of configuration, pay attention to security issues. For example, in most cases the Portal Server domain is located on the intranet, while the Secure Remote Access domain is in the DMZ.

Managing Risks

This section contains a few tips to help you in the sizing process.

- A business-to-consumer portal requires that you deploy Secure Remote Access to use the Gateway and SSL. Make sure you take this into account for your sizing requirements. Once you turn on SSL, the performance of the portal can be up to ten times slower than without SSL.
- For a business-to-employee portal, make sure that you have a user profile that serves as a baseline.
- For any portal, build in headroom for growth. This means not just sizing for today's needs, but future needs and capacity. This includes usual peaks after users return from a break, such as a weekend or holiday, or if usage is increased over time because the portal is more "sticky."
- If you are deploying your portal solution across multiple geographic sites, you need to fully understand the layout of your networks and data centers
- Decide what type of redundancy you need. Consider items such as production down time, upgrades, and maintenance work. In general, when you take a portal server out of production, the impact to your capacity should be no more than one quarter of the overall capacity.
- In general, usage concurrencies for a business-to-employee portal are higher than a business-to-consumer portal.

◆ ◆ ◆ CHAPTER 6

Implementing Your Deployment Design

During the implementation phase of the solution life cycle you work from specifications and plans created during deployment design to build and test the deployment architecture, ultimately rolling out the deployment into production.

Implementation is beyond the scope of this guide, however this chapter describes how to monitor and tune a deployment prototype.

This chapter contains the following sections:

- "About Implementing Deployment Designs" on page 101
- "Documenting the Portal" on page 102
- "Identity and Directory Structure Design" on page 105
- "Software Deployment" on page 109
- "Content and Design Implementation" on page 111

About Implementing Deployment Designs

After the deployment architecture has been approved and implementation specifications and plans have been completed, you enter the implementation phase of the solution life cycle. Implementation is a complex set of processes and procedures that requires careful planning to ensure success.

Implementation includes the following tasks:

- Building the network and hardware infrastructure
- Installing and configuring software according to an installation plan
- Migrating data from existing applications to the current solution
- Implementing a user management plan
- Designing and deploying pilots or prototypes in a test environment according to a test plan
- Designing and running functional tests and stress tests according to a test plan

- Rolling out the solution from a test environment to a production environment according to a rollout plan
- Training administrators and users of the deployment according to a training plan

Details of implementation are beyond the scope of this guide. However, the following sections provide overview information for some of these tasks.

Documenting the Portal

A comprehensive set of documentation on how your portal functions is an important mechanism to increasing the supportability of the system. The different areas that need to be documented to create a supportable solution include:

- System architecture
- Software installation and configuration
- Operational procedures, also known as a "run book"
- Software customizations
- Custom code
- Third-party products integration

The run book outlines troubleshooting techniques as well as the deployment life cycle. Make this book available during the training and transfer of knowledge phase of the project.

Tip – Do not wait until the end of the deployment project to begin this documentation phase. Documenting your portal should occur as an ongoing activity throughout the entire deployment.

Developing a Portal Prototype

▼ To Develop a Portal Prototype

1. **Identify and remove obvious bottlenecks in the processor, memory, network, and disk.**

2. **Setup a controlled environment to minimize the margin of error (defined as less than ten percent variation between identical runs).**

   By knowing the starting data measurement baseline, you can measure the differences in data performance between sample gathering runs. Be sure measurements are taken over an adequate period of time and that you are able to capture and evaluate the results of these tests.

Plan to have a dedicated machine for generating load simulation which is separate from the Portal Server machine. A dedicated machine helps you to uncover the origin of performance problems.

3 **Define a baseline performance for your deployment, before you add in the full complexity of the project.**

4 **Using this initial benchmark, define the transaction volume your organization is committed to supporting in the short term and in the long run.**

Determine whether your current physical infrastructure is capable of supporting the transaction volume requirement you have defined.

Identify services that are the first to max out as you increase the activity to the portal. This indicates the amount of headroom you have as well as identify where to expend your energies.

5 **Develop and refine the prototype workload that closely simulates the anticipated production environment agreed between you and the portal administrators and portal developers.**

6 **Measure and monitor your traffic regularly to verify your prototype.**

Track CPU utilization over time. Load usually comes in spikes and keeping ahead of spikes involves a careful assessment of availability capabilities.

Most organizations find that portal sites are "sticky" in nature. This means that site usage grows over time, even when the size of the user community is fixed, as users become more comfortable with the site. When the size of the user community also grows over time a successful portal site can see a substantial growth in the CPU requirements over a short period of time.

When monitoring a portal server's CPU utilization, determine the average web page latency during peak load and how that differs from the average latency.

Expect peak loads to be four to eight times higher than the average load, but over short periods of time.

7 **Use the model for long-range scenario planning. The prototype can help you understand how dramatically you need to change your deployment to meet your overall growth projections for upcoming years.**

8 **Keep the error logging level to `ERROR` and not `MESSAGE`. The `MESSAGE` error level is verbose and can cause the file system to quickly run out of disk space. The `ERROR` level logs all error conditions and exceptions.**

9 **Monitor customized portal applications such as portlets.**

10 **Monitor the following areas.**

- Portal Desktop

- Channel rendering time
- Sun Java™ System Access Manager
- Sun Java System Directory Server
- Sun Java System Virtual Machine
- Web container

The following sections explain issues in terms of portal performance variables and provides guidelines for determining portal efficiency.

Access Manager Cache and Sessions

The performance of a portal system is affected to a large extent by the cache hit ratio of the Access Manager cache. This cache is highly tunable, but a trade-off exists between memory used by this cache and the available memory in the rest of the heap.

You can enable the `amSDKStats` logs to monitor the number of active sessions on the server and the efficiency of the Directory Server cache. These logs are located by default in the `/var/opt/SUNWam/stats` directory. Use the `com.iplanet.am.stats.interval` parameter to set the logging interval. Do not use a value less than five (5) seconds. Values of 30 to 60 seconds give good output without impacting performance.

The `com.iplanet.services.stats.directory` parameter specifies the log location, whether to a file or to the Portal Server administration console, and also is used to turn off the logs. You must restart the server for changes to take effect. Logs are not created until the system detects activity.

Note – Multiple web container instances write logs to the same file.

The cache hit ratio displayed in the `amSDKStats` file gives both an internal value and an overall value since the server was started. Once a user logs in, the user's session information remains in cache indefinitely or until the cache is filled up. When the cache is full, oldest entries are removed first. If the server has not needed to remove a user's entry, it might be the case that on a subsequent login—days later, for example—the user's information is retrieved from the cache. Much better performance occurs with high hit ratios. A hit ratio of a minimum of 80 percent is a good target although (if possible) an even higher ratio is desired.

Thread Usage

Use the web container tools to monitor the number of threads being used to service requests. In general, the number of threads actually used is generally lower than many estimates, especially in production sites where CPU utilization usually is far less than 100 percent.

Portal Usage Information

Portal Server does include a built-in reporting mechanism to monitor portal usage information by portal users. This includes which channels are accessed, how long the channels are accessed, and the ability to build a user behavioral pattern of the portal. Portal monitoring can be administered using the psconsole or cli psadmin. The following monitoring capabilities are possible:

- Managed Desktop Services:
  - Content Request
  - Edit Request
  - Process Request
  - Logout Request
- Services exposed
  - cumulatively
  - per channel
- Each service has set of attributes
- Service Attributes:
  - Name Count
  - Last Service Time
  - Maximum Service Time
  - Minimum Service Time
  - Average Service Time

User Based Tracking (UBT) is also administered using the `psconsole` or cli `psadmin`. UBT is disabled by default. To enable user based tracking, set property `com.sun.portal.ubt.enable=true` in file `/var/opt/SUNWportal/portals/portal1/config`. The level `INFO`, `FINE`, `FINER`, `FINEST`, `OFF` controls extent of data captured. Logs are routed to a file: `/var/opt/SUNWportal/portals/portal1/logs/%instance/ubt.%u.%g.log` which is configurable.

Identity and Directory Structure Design

A major part of implementing your portal involves designing your directory information tree (DIT). The DIT organizes your users, organizations, suborganizations into a logical or hierarchical structure that enables you to efficiently administer and assign appropriate access to users.

The top of the organization tree in Access Manager is called `dc=`*fully-qualified-domain-name* by default, but can be changed or specified at install time. Additional organizations can be created after installation to manage separate enterprises. All created organizations fall beneath the top-level organization. Within these suborganizations other suborganizations can be nested. The depth of the nested structure is not limited.

Note – The top of the tree does not have to be called `dc`. Your organization can change this to fit its needs. However, when a tree is organized with a generic top, for example, `dc`, then organizations within the tree can share roles.

Roles are a grouping mechanism designed to be more efficient and easier to use for applications. Each role has members, or entries that possess the role. As with groups, you can specify role members either explicitly or dynamically.

The roles mechanism automatically generates the `nsRole` attribute containing the distinguished name (DN) of all role definitions in which the entry is a member. Each role contains a privilege or set of privileges that can be granted to a user or users. Multiple roles can be assigned to a single user.

The privileges for a role are defined in Access Control Instructions (ACIs). Portal Server includes several predefined roles. The Portal Server administration console enables you to edit a role's ACI to assign access privileges within the Directory Information Tree. Built-in examples include `SuperAdmin Role` and `TopLevelHelpDeskAdmin` roles. You can create other roles that can be shared across organizations.

Creating a Custom Access Manager Service

Service Management in Access Manager provides a mechanism for you to define, integrate, and manage groups of attributes as an Access Manager service.

Readying a service for management involves:

- Creating an XML service file
- Configuring an LDIF file with any new object classes and importing both the XML service file and the new LDIF schema into Directory Service
- Registering multiple services to organizations or sub-organizations using the Access Manager administration console
- Managing and customizing the attributes (once registered) on a per organization basis

See the *Sun Java System Portal Server 6 Secure Remote Access 2005Q4 Administration Guide*, *Sun Java System Directory Server Enterprise Edition 6 2006Q1 Deployment Planning Guide* and the *Access Manager Deployment Guide* for more information on planning your Access Manager and Directory Server structure.

Portal Desktop Design

The performance of Portal Server largely depends upon how fast individual channels perform. In addition, the user experience of the portal is based upon the speed with which the Portal Desktop is displayed. The Portal Desktop can only load as fast as the slowest displayed channel. For example, consider a Portal Desktop composed of ten channels. If nine channels are rendered in one millisecond but the tenth takes three seconds, the Portal Desktop does not appear until that tenth channel is processed by the portal. By making sure that each channel can process a request in the shortest possible time, you provide a better performing Portal Desktop.

Choosing and Implementing the Correct Aggregration Strategy

The options for implementing portal channels for speed and scalability include the following.

- Keeping processing functions on back-end systems and application servers, not on the portal server. The portal server needs to optimize getting requests from the user. Push as much business logic processing to the back-end systems. Whenever possible, use the portal to deliver customized content to the users, not to process it.
- Ensuring that the back-end systems are highly scalable and performing. The Portal Desktop only responds as fast as the servers from which it obtains information (to be displayed in the channels).
- Understanding where data is stored when designing providers, how the portal gets that data, how the provider gets that data, and the type of data. For example, is the data dynamic that pertains to an individual user, or is there code needed to retrieve that customized or personalized data? Or, is the data static and shared by a small group of users? Next, you need to understand where the data resides (for example, in an XML file, database and flat file), and how frequently the data is updated. Finally, you need to understand how the business logic is applied for processing the data, so that the provider can deliver a personalized channel to the user.

Working with Providers

Consider the following when planning to deploy providers:

- **URLScraperProvider**. Typically you use this provider to access dynamic content that is supplied by another web container's web-based system. It uses HTTP and HTTPS calls to retrieve the content. This provider puts high requirements on the back-end system, as the back-end system has to be highly scalable and available. Performance needs to be in tenths of a second to show high performance. This provider is very useful for proof of concept in the trial phase of your portal deployment due to the simplicity of configuration.

  `URLScraperProvider` also performs some level of rewriting every time it retrieves a page. For example, if a channel retrieves a news page that contains a picture that is hosted on another web site, for the portal to be able to display that picture, the URL of that picture needs to be rewritten. The portal does not host that picture, so `URLScraperProvider` needs to rewrite that picture URL to present it to portal users.

  The URL Scraper provider that is part of Portal Server can also function as a file scraper provider.

  To use URLScraperProvider as a file scraper provider, specify the URL as follows:

  `String name="url"value="file://`*path/filename*"

  This is the best performing provider, in terms of how fast it retrieves content. On the first fetch of content, performance for this provider is usually in the low ten milliseconds. On subsequent requests, using a built-in caching mechanism, this provider can usually deliver content in one millisecond or less. If applicable, consider using the file scraper provider in place of the URL Scraper provider.
- **JSPProvider.** Uses JavaServer Pages™ (JSP) technology. JSPProvider obtains content from one or more JSP files. A JSP file can be a static document (HTML only) or a standard JSP file with HTML and Java programming language embedded within tags. A JSP file can include other JSP files. However, only the topmost JSP file can be configured through the display profile. The topmost JSP files are defined through the `contentPage`, `editPage`, and `processPage` properties.
- **LoginProvider.** Provides access to the Access Manager authentication service through a Portal Desktop channel. This provider enables anonymous Portal Desktop login so that a user can log in directly from the Portal Desktop.
- **XMLProvider.** Transforms an XML document into HTML using an XSLT (XML Style Sheet Language) file. You must create the appropriate XSLT file to match the XML document type. `XMLProvider` is an extension of `URLScraperProvider`. This provider uses the `javax.xml.transform` classes provided with j2se 1.5.
- **SimpleWebServiceProvider.**As a subclass of `URLScraperProvider`, `SimpleWebServicProvider` retrieves the WSDL file and communicates with the associated Web service through Simple Object Access Protocol (SOAP).

Software Deployment

This section provides information on the software packaging mechanism, the software categories within the system, and compatibility with Java software.

Software Packaging

Portal Server uses a "dynamic WAR file" approach to deploy software to the system. Portal Server is installed using Solaris™ packages, which consist of individual files that comprise web applications, for example, JAR, JSP, template, and HTML files. The packages do not contain WAR or EAR files. The packages do contain `web.xml` fragments that are used to construct the Portal Server WAR file at installation time. This dynamically constructed file is then deployed to the web application container. As additional packages are added to the system, for example, for localization, the web application file is rebuilt and redeployed.

Note – The WAR file packaging and deployment mechanism is for use only by Portal Server products. Customer modifications to the WAR file or any files used to build it are currently not supported.

Software Categories

Portal Server distinguishes between the following kinds of software that it installs onto the Portal Server node:

- **Dynamic web applications.** These include servlets running on a Java platform, JSP files, content providers, and other items that the web container processes when accessed by the user's browser. For Portal Server, these files are deployed in the web container directory structure.
- **Static web content.** These include static HTML files, images, applet JAR files, and other items that can be served up directly by the web server without using the Web Server container. For Portal Server, these files are also deployed in the web container directory structure.

Note – Static web content and dynamic web applications are all grouped together into a single WAR file.

- **Configuration data.** These include data that is installed into the directory, that is, the Access Manager service definitions and any other data that modifies the directory at installation time. This includes modifications to the console configuration data to connect in the Portal Server extensions. Configuration data is installed only once no matter how many Portal Server nodes there are.

- **Software Development Kit (SDK).** This is the JAR file or files that contain the Java APIs that are made available by a component. Developers need to install this package on a development system so that they can compile classes that use the API. The portal sdk files are located under the directory `/opt/SUNWportal/sdk`. The desktop subdirectory contains the `desktopsdk.jar` file required when building custom providers. The wsrp sub-directory contains the `wsrpsdk.jar` file.

Compatibility With Java Software

Portal Server software falls into three categories:

- Applets. Applets used in Portal Server are compatible with Java 1.1, which is supported by most browsers.
- Web applications. Web applications are intended to be compatible with the (J2EE™) web container based on the servlet's interface except where uses of special interfaces are identified. Web application compatibility includes Java 2 (J2SE 1.3) and later.
- Stand-alone Java processes. Stand-alone Java software processes are compatible with Java 2 and later. Some Portal Server software, specifically in Secure Remote Access, use Java Native Interface (JNI) to call C application programming interfaces (APIs).

Client Support

Portal Server supports the following browsers as clients:

- Internet Explorer 5.5 and 6.0
- Netscape™ Communicator 4.8or higher
- Netscape 7.1 or higher
- Mozilla 1.7.12 or higher
- Firefox 1.0.7 or higher

Multiple client types, whether based on HTML, WML, or other protocols, can access Access Manager and hence Portal Server. For this functionality to work, Access Manager uses the Client Detection service (client detection API) to detect the client type that is accessing the portal. The client type is then used to select the portal template and JSP files and the character encoding that is used for output.

Note – Currently, Access Manager defines client data only for supported HTML client browsers, including Internet Explorer and Netscape Communicator. See the Access Manager documentation for more information.

Sun Java System Portal Server Mobile Access software extends the services and capabilities of the Portal Server platform to mobile devices and provides a framework for voice access. The software enables portal site users to obtain the same content that they access using HTML browsers.

Mobile Access software supports mobile markup languages, including xHTML, cHTML, HDML, HTML, and WML. It can support any mobile device that is connected to a wireless network through a LAN or WAN using either the HTTP or HTTPS protocol. In fact, the Portal Server Mobile Access software could support any number of devices, including automobiles, set-top boxes, PDAs, cellular phones, and voice.

Content and Design Implementation

The Portal Desktop provides the primary end-user interface for Portal Server and a mechanism for extensible content aggregation through the Provider Application Programming Interface (PAPI). The Portal Desktop includes a variety of providers that enable container hierarchy and the basic building blocks for building some types of channels. For storing content provider and channel data, the Portal Desktop implements a display profile data storage mechanism on top of an Access Manager service.

The various techniques you can use for content aggregation include:

- Creating channels using building block providers
- Creating channels using `JSPProvider>`
- Creating channels using Portal Server tag libraries
- Creating channels using custom building block providers
- Organizing content using container channels

See the *Sun Java System Portal Server 7.1 Secure Remote Access Administration Guide* and *Sun Java System Portal Server 7.1 Developer Sample Guide* for more information.

Placement of Static Portal Content

Place your static portal content in the *web-container-instance-root/* `https-server/docs` directory or in a subdirectory under the *web-container-instance-root/*`directory/https-server/docs` (the document root for the web container). Do not place your content in the *web-container-instance-root/*`SUNWportal/web-apps/https-server/portal/` directory, as this is a private directory. Any content here is subject to deletion when the Portal Server web application is redeployed during a patch or other update.

Integrating Applications

Integrating and deploying applications with Portal Server is one of your most important deployment tasks. The application types include:

- **Channel.** Provides limited content options; is not a “mini-browser”.
- Portlet. Pluggable web component that processes requests and generates content within the context of a portal. In Portal Server software, a portlet is managed by the Portlet Container. Conceptually, a portlet is equivalent to a Provider.
- **Portal application.** Launched from a channel in its own browser window; the Portal Server hosts the application; created as an Access Manager service; accesses Portal and Access Manager APIs.
- **Third-party application.** Hosted separately from Portal Server, but accessed from Portal Server; URL Scraper, which calls Rewriter, rewrites web pages so that the web pages can be displayed in a channel; uses Access Manager to enable single sign-on.

Integrating Microsoft Exchange/Sun Java System Messaging Server

Using the JavaMail™ API is one of the primary options for integrating Microsoft Exchange messaging server with Portal Server. The JavaMail API provides a platform independent and protocol independent framework to build Java technology-based mail and messaging applications. The JavaMail API is implemented as a Java optional package that can be used on JDK 1.4 and later on any operating system. The JavaMail API is also a required part of the Java Platform, Enterprise Edition (Java EE).

JavaMail provides a common uniform API for managing mail. It enables service providers to provide a standard interface to their standards based or proprietary messaging systems using Java programming language. Using this API, applications can access message stores and compose and send messages.

Independent Software Vendors

Listed below are some types of independent software vendor (ISV) integrations.

- **Application user interface.**. This integration uses the provider API and Secure Remote Access for secure access. (Secure Remote Access is not an integration type on its own.) Examples include FatWire, Interwoven, SAP, Tarantella, Documentum, Vignette, PeopleSoft, Siebel, Citrix, and YellowBrix.
- **Security products.**. This integration uses the Access Manager Login API to enable portal access by using a custom authentication scheme. Examples include RSA Security.
- **Content Management.** This integration provides data access into Portal Server, enabling searches on the data. Examples include FatWire, Interwoven, and Vignette.
- **Content Syndication.** This integration provides managing and customizing information that appears on websites. Examples include YellowBrix and Pinnacor.

- **Collaboration software.** This integration enables Sun Java System InstantMessaging product to move a collaboration session from one forum to a another. Examples include WebEx, BeNotified, and Lotus.
- **Monitoring.** This integration focuses on billing, performance measurement, and diagnostics, for which you rely on log files (or Portal Server's Logging API) and traffic snooping. Examples include Mercury Interactive, Hyperion, and Informatica.
- **Portal capability augmentation.** This integration enables products to add functionality to Portal Server. Examples include Altio, Bowstreet, rule engines to add group capability, and dynamic standard Portal Desktop and provider contents (HNC).
- **Integratable portal stack.** This integration includes products that replace elements of Portal Server. Examples include Access Manager and LDAP.

Note – Portal Server 7.1 is reliant on using the `amsdk`. Limited Realm Mode support is possible, however by default portal 7.1 will install in legacy mode. Sun One Java Directory Server 6 is supported and any LDAPv3 directory server.

The "depth" to which user interface integration occurs with Portal Server indicates how complete the integration is. Depth is a term used to describe the complementary nature of the integration, and points to such items as:

- Application availability through Portal Server
- Application availability in secure mode (using Secure Remote Access, Netlet rules)
- Ability to use single sign-on

In general, the degree to which an application integrates in Portal Server can be viewed as follows:

- **Shallow integration**. This integration essentially uses the Portal Server as a launch point. The user logs in to the portal and clicks a link that starts a web application.
- **Deep integration**. The user accesses the user interface provided by the channels in Portal Server directly. That is, the integrated software works within the portal. No additional windows or applets appear.

Rolling Out a Production Deployment

Once the pilot or proof-of-concept deployment passes the test criteria, you are ready to roll out the deployment to a production environment. Typically, you roll out to a production environment in stages. A staged rollout is especially important for large deployments that affect a significant number of users.

The staged deployment can start with a small set of users and eventually expand the user base until the deployment is available to all users. A staged deployment can also start with a limited set of services and eventually phase in the remaining services. Staging services in phases can help isolate, identify, and resolve problems a service might encounter in a production environment.

CHAPTER 7

Deploying Portal Server Communities

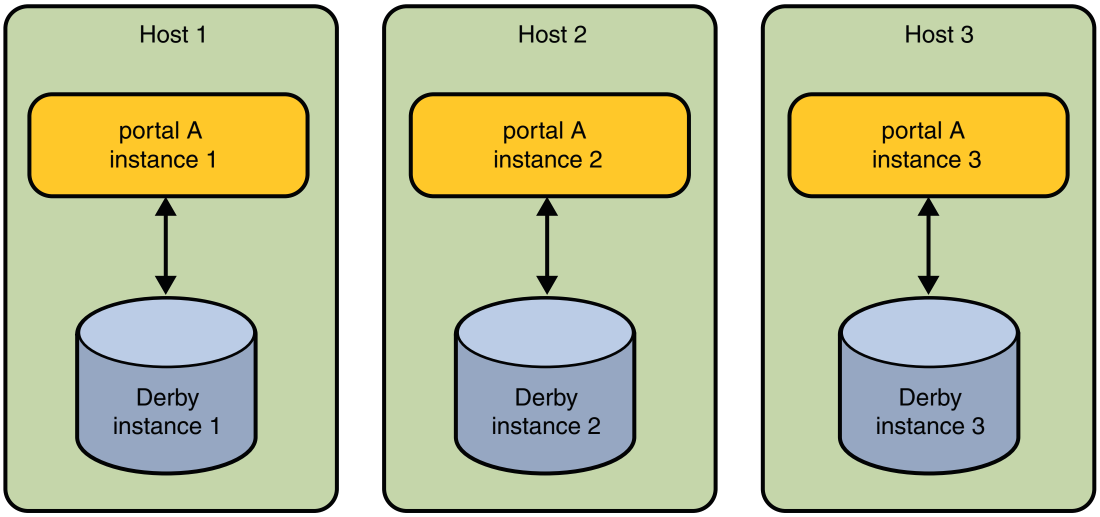
Portal Server communities are associations of members and services that end users can establish and manage themselves. All the community data (community membership and configuration data, as well as application data for portlets) is designed to be stored in a database of the administrator's choice that can be connected to using a jdbc driver. Portal 7.1 uses Derby database by default for storing the above data and is shipped with the Portal product. As such it becomes important to consider tuning this database for performance and scalability.

This chapter provides the following common examples of deployment scenarios for Portal Server communities.

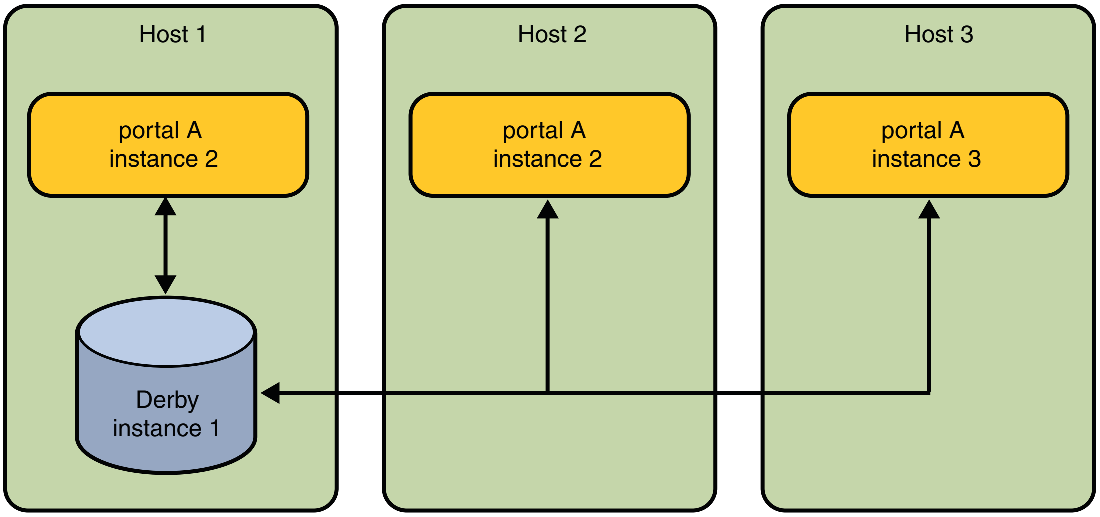
- "Deploying Communities on the Default Database Installation" on page 115
- "Deploying Communities on Multiple Hosts Using a Single Database Instance" on page 117
- "Deploying Communities on a High-Availability Database" on page 124

Deploying Communities on the Default Database Installation

By default, each Portal Server installation creates a instance of a Derby Database. This instance is configured as a Network Server that is accessible from any client using the Derby Client JDBC.



When multiple Portal instances are deployed to different hosts, but serving the same community, it is necessary to use a common Derby server instance.



To deploy Portal communities on multiple hosts that use a single database instance, see "Deploying Communities on Multiple Hosts Using a Single Database Instance" on page 117.

By default, the Derby Network Server is configured to listen on port 1527. To test if derby is running use the following UNIX command

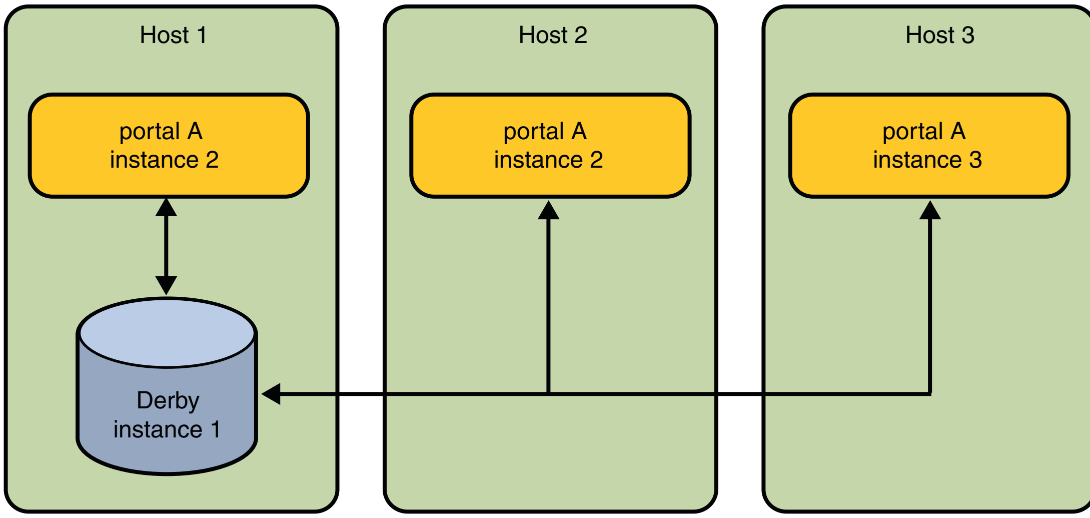
```
netstat -an | grep 1527
```

If there is nothing listening on the port the derby database is not running. It can be started using the following lines:

```
###start_derby.sh####
PATH=/usr/jdk/jdk1.5.0_09/bin:${PATH};export PATH
derby_classpath=/opt/SUNWjavadb/lib/derby.jar:/opt/SUNWjavadb/lib/derbynet.jar;
exportderby_classpath
derby_home=/var/opt/SUNWportal/derby;export derby_home
java -Dderby.system.home=${derby_home} -cp ${derby_classpath}
org.apache.derby.drda.NetworkServerControl start &
```

Deploying Communities on Multiple Hosts Using a Single Database Instance

When multiple Portal instances are deployed to different hosts, each instance has different community data unless you configure the instances to use a common database instance. This section describes how to configure multiple Portal Server instances on different hosts to use a single, common Derby network server instance.



Deploying multiple instances of Portal Server on a single, common Derby instance can be accomplished in two ways.

- You can use a Derby instance that is already installed on a Portal Server host and configure other Portal Server instances to point to the Derby instance. "To Point All Portal Server Instances To The Common Derby Network Server Instance" on page 120
- You can create a derby instance on a dedicated Derby host and configure other Portal Server instances to point to the host running the dedicated Derby instance.

1. “To Set Up a Common Derby Network Server Instance (Solaris)” on page 118
2. “To Point All Portal Server Instances To The Common Derby Network Server Instance” on page 120

Deployment Requirements

The platform requirements for deploying multiple Portal Server communities on single Derby instances include the following:

Operating System

- Solaris 9 U6 or Solaris 10 on SPARC
- Solaris 9 or Solaris 10 on x86
- RedHat Enterprise Linux 2.1, 3.0 Update 3 or 4.0 on x86.

For more detailed information please refer to the release notes.

These requirements are similar to those required for setting up any portal instance. The requirements for setting up a dedicated derby instance also require the use of java sdk which is normally packaged with the Java Enterprise System 5. The required version is 1.5.0_09 or later.

▼ To Set Up a Common Derby Network Server Instance (Solaris)

1 Install JDK from Java Enterprise System 5 distribution. This step is required if the required Java version is not present on the machine.

a.
```
cd
/net/machine1.pstest.com/jes5/Solaris_sparc/Product/shared_components/Packages
```

b.
```
pkgadd -d . SUNWj5rt SUNWj5rtx SUNWj5cfg SUNWj5dev SUNWj5dmo SUNWj5dmx
SUNWj5dvx SUNWj5man
```

2 Install the Derby database packages on the dedicated host.

a.
```
cd
/net/machine1.pstest.com/jes5/Solaris_sparc/Product/shared_components/Packages
```

b.
```
pkgadd -d . SUNWjavadb-common SUNWjavadb-core SUNWjavadb-client
SUNWjavadb-demo SUNWjavadb-docs SUNWjavadb-javadoc
```

3 **Transfer the Derby data files from the Portal Server install host to the dedicated Derby network server instance host. The data files are located in `/var/opt/SUNWportal/derby`.**

The Derby data files are:

- derby.properties
- and the following directories
  - communitymc_portal1
  - surveydb_portal1
  - filesharingdb_portal1
  - wikidb_portal1

a. **Archive the Derby data files using the tar command on the Portal Server install host.**

```
cd /var/opt/SUNWportal/derby

tar -cvf derby-system.tar *
```

b. **Create the Derby system home directory on the dedicated Derby network instance host.**

```
mkdir -p /var/opt/SUNWportal/derby
```

c. **Transfer the `derby-system.tar` file from the Portal Server install host to the system home directory on the dedicated Derby network instance host.**

On the dedicated Derby network instance host, execute the following commands from the `/var/opt/SUNWportal/derby` directory.

```
ftp> portal install

ftp> cd /var/opt/SUNWportal/derby

ftp> get derby-system.tar

ftp> quit
```

d. **Extract the `derby-system.tar` file using the tar command.**

```
tar -xvf derby-system.tar
```

4 **Modify the Derby properties file.**

a. **Change the `derby.drda.host` property to the fully qualified host name assigned to the interface on which the common Derby network server instance runs.**

b. **Delete the derby log file**

```
rm derby.log
```

c. **create empty log file.**

```
touch derby.log
```

d. **(Optional) Change the `derby.drda.port` property to run the server on a non default port. Port 1527 is the Derby default port. Changing the port property is optional.**

5 **Start the Derby network server instance. You can use the following sample script:**

```
###start_derby.sh####
PATH=/usr/jdk/jdk1.5.0_09/bin:${PATH};export PATH
derby_classpath=/opt/SUNWjavadb/lib/derby.jar:/opt/SUNWjavadb/lib/derbynet.jar;
exportderby_classpath
derby_home=/var/opt/SUNWportal/derby;export derby_home
java -Dderby.system.home=${derby_home} -cp ${derby_classpath}
org.apache.derby.drda.NetworkServerControl start &
```

6 **(Optional) Modify the Derby tuning properties in `derby.properties` or add JVM tuning parameters to the Java command used to start the Derby network server. For details, refer to the Apache Derby manuals, and java.sun.com.**

▼ To Point All Portal Server Instances To The Common Derby Network Server Instance

Portal clients of the Derby network server instance access it through a JDBC data source that is configured in the web container. Configuring a Portal instance to use a non-default Derby network server instance means reconfiguring these data sources to point to a different host. The method depends on the brand of web container. This procedure uses Sun Java Webserver 7.0 as an example.

Perform the following procedure for each Portal Server instance that you want to use a common Derby network server instance.

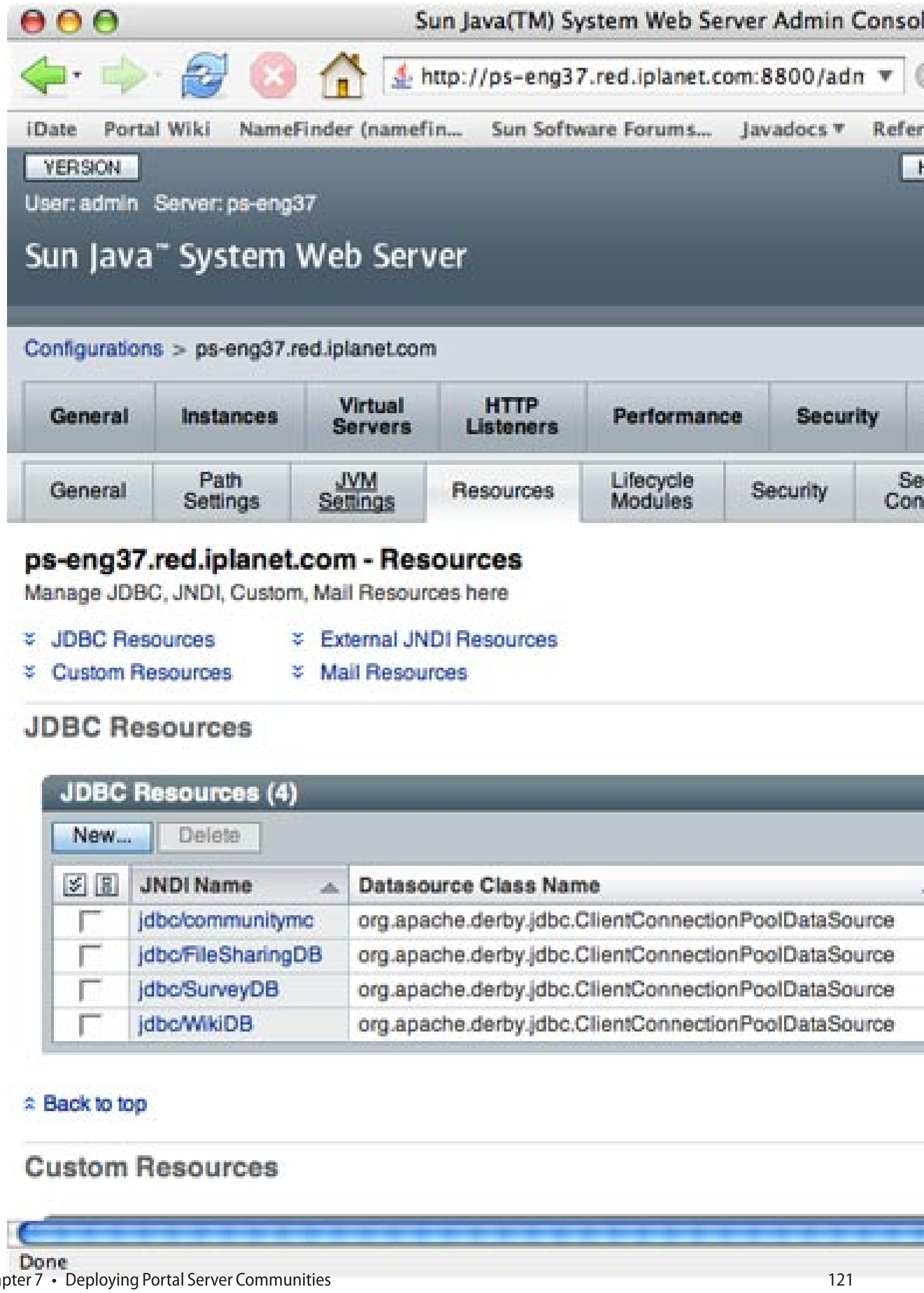
1 **Access the Sun Java System Web Server.**

For example, http://sesta.iplanet.com:8800

2 **Under the Common Tasks Tab, click Edit Java Settings under Configuration Tasks.**

3 **Click the Resources tab.**

A list of the JDBC resources in the web container is displayed.



ps-eng37.red.iplanet.com - Resources

Manage JDBC, JNDI, Custom, Mail Resources here

- JDBC Resources
- External JNDI Resources
- Custom Resources
- Mail Resources

JDBC Resources

JDBC Resources (4)

New... Delete

	JNDI Name	Datasource Class Name
	jdbc/communitymc	org.apache.derby.jdbc.ClientConnectionPoolDataSource
	jdbc/FileSharingDB	org.apache.derby.jdbc.ClientConnectionPoolDataSource
	jdbc/SurveyDB	org.apache.derby.jdbc.ClientConnectionPoolDataSource
	jdbc/WikiDB	org.apache.derby.jdbc.ClientConnectionPoolDataSource

Back to top

Custom Resources

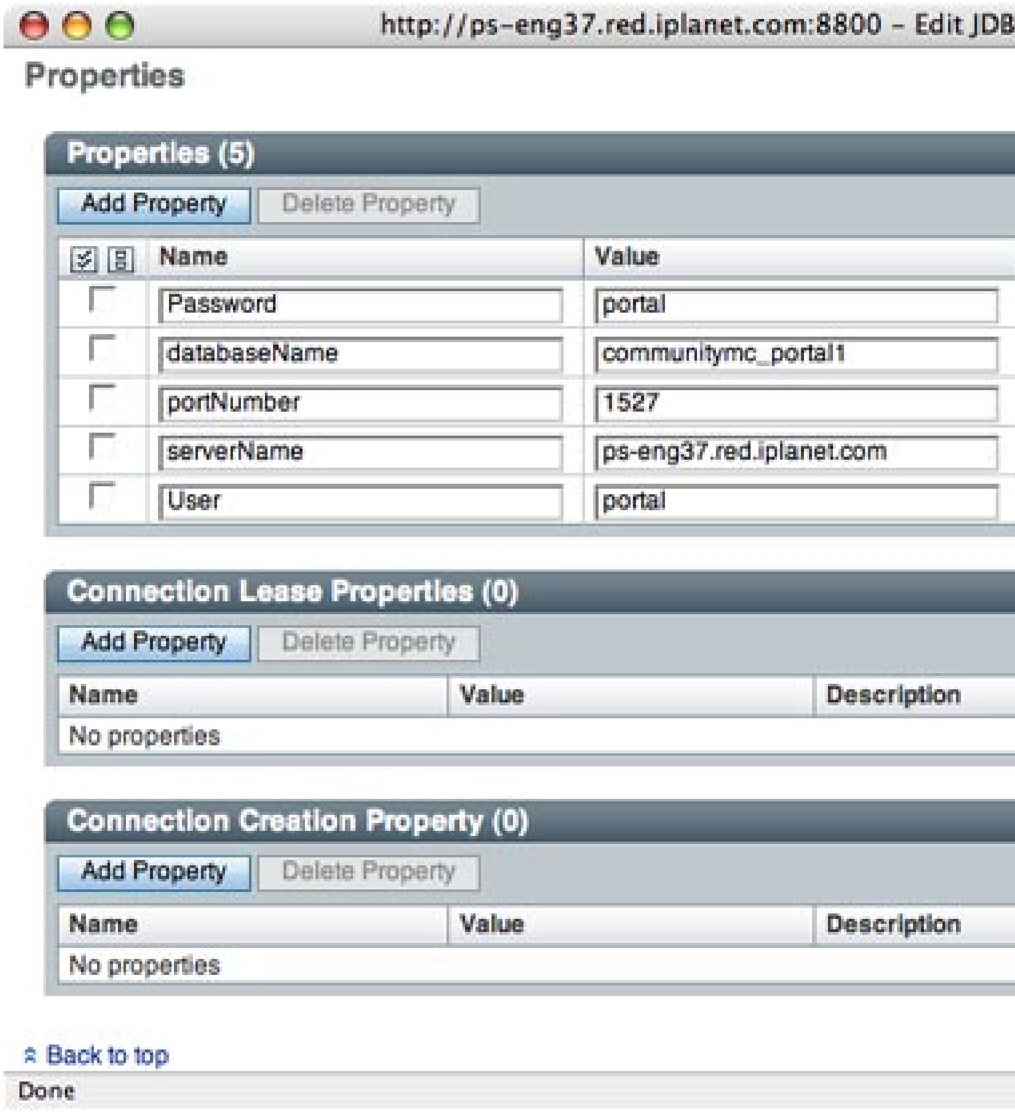
4 Perform the following steps for each of the following resources:

- `jdbc/communitymc`
- `jdbc/FileSharingDB`
- `jdbc/SurveyDB`
- `jdbc/WikiDB`

a. Click the resource.

A new window opens.

b. Click the Properties link at the top of the window to scroll to the resource properties.



http://ps-eng37.red.iplanet.com:8800 – Edit JDB

Properties

Properties (5)

Add Property | Delete Property

Name	Value
Password	portal
databaseName	communitymc_portal1
portNumber	1527
serverName	ps-eng37.red.iplanet.com
User	portal

Connection Lease Properties (0)

Add Property | Delete Property

Name	Value	Description
No properties		

Connection Creation Property (0)

Add Property | Delete Property

Name	Value	Description
No properties		

Back to top

Done

c. **Change the `serverName` property to the host name of the common Derby server instance.**

d. **Click Ok.**

5 **Deploy the configuration by performing the following steps:**

a. **Select the Configurations Tab.**

b. Identify the configuration by selecting the configuration checkbox.

c. Select Deploy Configuration from the configuration action list.

6 Restart the Web Container.

Deploying Communities on a High-Availability Database

To provide high availability to an enterprise scale community, the Derby database that is installed with Portal Server must be replaced by a jdbc connectable database that provides high availability.

▼ To Replace the Derby Database

1 Set up the database.

a. Install RDBMS or identify one that already exits on the system.

b. Create Database instance (or tablespace in the case of Oracle) to be used by collaboration.

c. Create database user accounts.

d. Establish appropriate privileges for the user accounts.

2 Set up the Web Container for the New Database.

a. Locate the JDBC driver.

b. Add the JDBC driver to the web container's JVM classpath.

c. Add the JVM option:

-Djdbc.drivers=*JDBC_DRIVER_CLASS*

3 Configure Community Membership and Configuration.

a. Configure the communitymc database configuration file:

PORTAL-DATA-DIR`/portals/`*PORTAL-ID*`/config/portal.dbadmin`.

Parameters

`db.driver=`*vendor-specific-classpath*`.jdbc.ClientDriver`

`db.driver.classpath=/opt/`*jarfile-directory-path*`client.jar` `community.db.user=`*database-user-account* (created in step 1c).

`community.db.password=`*database-user-account-password* (created in step 1c).

`community.db.url=jdbc:xxx://xxxxx.pstest.com:`*port*/*database-instance-created-in-step-1b*.

b. **Remove the derby-specific property in:**
PORTAL-DATA-DIR`/portals/`*PORTAL-ID*`/config/communitymc.properties`.

For example: `javax.jdo.option.Mapping=derby`

c. **Load the schema onto the database by using the following commands:**

```
cd /var/opt/SUNWportal/portals/portal1/config
```

```
/usr/sfw/bin/ant -f config.xml -D"portal.id=portal1" configure
```

d. **Edit the JDBC resource in the web container configuration to point to the new database.**

4 **Configure and Install Portlet Applications.**

a. **Locate the portlet applications:**
PORTAL-DATA-DIR`/portals/`*PORTAL-ID*`/portletapps`

b. **Configure the portlet applications to use the new database, for example:**

```
cd /var/opt/SUNWportal/portals/portal1/portletapps/surveys
```

Edit the `tokens_xxx.properties` file.

c. **Create the JDBC Resource for each application using the values from the** `tokens_xxx.properties` **file.**

- Resource JNDI Name: `jdbc/`*DB-JNDI-NAME*
- Resource Type: javax.sql.DataSource
- Datasource Classname: *DB-DATASOURCE*
- User: *DB-USER*
- Password: *DB-PASSWORD*
- URL: *DB-URL*

d. **Undeploy the existing portlets that use Derby Database as the datastore.**

e. **Deploy the newly configured portlet applications.**

▼ To Deploy High-Availability Database

Before You Begin Install the Oracle 10g Release 2 database.

1. **Prepare the Oracle database.**

   a. **Create a database instance named** `portal`**. An example is,** `SID=portal`**.**

   Note – To avoid ORA-27102 out of memory errors on Solaris 10, perform the following command: `projadd -U oracle -K "project.max-shm-memory=(priv,4294967295B,deny)" user.oracle`

   ---

   b. **b. Log in to the Oracle Enterprise Manager as SYSTEM. The path is typically** `http://hostname:1158/em`**. (em stands for Enterprise Manager).**

   or command line `sqlplus /nolog CONNECT sys/password AS SYSDBA`;

   c. **Create a tablespace** `communitymc_`*portal-ID* **for example,** `communitymc_portal1`**.**

   For example,

   ```
   SQL> CREATE TABLESPACE COMMUNITYMC_PORTAL1 LOGGING DATAFILE ’
   /app/oracle/oradata/portal/communitymc_01.dbf’ SIZE 400M REUSE AUTOEXTEND ON NEXT
   1280K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL;
   ```

   d. **Create a user account with the following values:**

   Username: `portal`

   Password: `portal`

   Default Tablespace: `communitymc_`*Portal—ID*

   Assign roles: CONNECT and RESOURCE

   For example,

   ```
   SQL> create user portal identified by portal;

   SQL>alter user portal default tablespace  COMMUNITYMC_PORTAL1;

   SQL> grant connect,resource to portal;
   ```

2. **Prepare the web container for the new database.**

   a. **Locate the Oracle JDBC driver using one of the following methods. The file is**`ojdbc14.jar`**.**

   - Use the existing `ojdbc14.jar` file on the machine on which Oracle is installed. It is normally located in: `$ORACLE_HOME/jdbc/lib/ojdbc14.jar`.

- Download the `ojdbc14.jar` file from the Oracle website. Verify that you download the version that is compatible with the your Oracle RDBMS.

b. Add the JDBC driver to the JVM classpath.

Add the JDBC driver (`ojdbc14.jar`) to the JVM classpath.

For example,

class-path-suffix`...//opt/SUNWjavadb/lib/derbyclient.jar:`
`/oracle/ojdbc14.jar...`

c. Add the following JVM option:

`-Djdbc.drivers=oracle.jdbc.OracleDriver`.

For example, in Web Server server.xml

```
<jvm-options>-Djdbc.drivers=org.apache.derby.jdbc.ClientDriver</jvm-options>
```

```
<jvm-options>-Djdbc.drivers=oracle.jdbc.OracleDriver</jvm-options>
```

Note – It is possible to remove the derby classfiles and options once all the portlets/applications that are using derby database jdbc resources have been converted to using oracle resources.

---

3 Configure Community Membership and Configuration.

a. Edit the `communitymc` database configuration file:

vi *Portal-data-dir*`/portals/`*Portal-ID*`/config/portal.dbadmin`

`db.driver=oracle.jdbc.OracleDriver`

`db.driver.classpath=`*JDBC-Driver-Path*`/ojdbc14.jar`
`url=jdbc:oracle:thin:@`*Oracle-Host*`:`*Oracle-Port*`:portal`

For example, `portal.dbadmin`

```
db.driver=oracle.jdbc.OracleDriver
db.driver.classpath=/oracle/ojdbc14.jar
community.db.user=portal
community.db.password=portal
community.db.url=jdbc:oracle:thin:@machine1.pstest.com:1521:portal
portal.id=portal1
```

b. Remove or comment out the following property from the `communitymc` configuration file.

`vi` *Portal-Data-Dir*`/portals/`*Portal-ID*`/config/communitymc.properties`

```
#javax.jdo.option.Mapping=derby
```

c. **Load the community schema onto Oracle by using the following commands:**

```
cd /var/opt/SUNWportal/portals/portal1/config
```

```
/usr/sfw/bin/ant -f config.xml -D"portal.id=portal1" configure
```

d. **When finished check that tables have been created in oracle by typing the following:**

```
SQL> connect portal/portal
Connected.
SQL> select * from cat;

TABLE_NAME                     TABLE_TYPE
------------------------------ -----------
BIN$I3usV+V+I7HgRAgAIMbswQ==$0 TABLE
JPOX_TABLES                    TABLE
COMMUNITY                      TABLE
COMMUNITY_USER                 TABLE
COMMUNITY_DP                   TABLE
```

e. **Edit the `communitymc` JDBC resource to point to Oracle.**

i. **Log in to the web container administration console.**

ii. **Locate the JDBC resource named `jdbc/communitymc`.**

iii. **Set the Datasource classname to `oracle.jdbc.pool.OracleDataSource`.**

iv. **Set following properties:**

- user: `portal`
- password: `portal`
- url: `jdbc:oracle:thin:@`*Oracle-Host*:*Oracle-Port*:portal

```
<jdbc-resource>
    <jndi-name>jdbc/communitymc</jndi-name>
    <datasource-class>oracle.jdbc.pool.OracleDataSource</datasource-class>
    <idle-timeout>30</idle-timeout>
    <wait-timeout>10</wait-timeout>
    <property>
      <name>Password</name>
      <value>portal</value>
    </property>
    <property>
      <name>portNumber</name>
      <value>1521</value>
    </property>
    <property>
      <name>databaseName</name>
      <value>communitymc_portal1</value>
    </property>
    <property>
      <name>User</name>
      <value>portal</value>
    </property>
    <property>
      <name>serverName</name>
      <value>machine1.pstest.com</value>
      <description/>
    </property>
    <property>
      <name>url</name>
      <value>jdbc:oracle:thin:@machine1.pstest.com:1521:portal</value>
      <description/>
    </property>
  </jdbc-resource>
```

Note – On some web containers, you might need to edit the JDBC Connection Pool instead of the JDBC resource.

4 Configure and Install the Fileshare, Surveys, and Wiki Portlet Applications.

a. Locate the portlet application:

Portal-Data-Dir/portals/*Portal-ID*/portletapps/*Portlet_Application*

b. Configure `tokens_ora.properties`**.**

Each application located under `/portals/portal1/portletapps/` has its own `tokens_ora.properties` file in addition to a `tokens.properties` file. The necessary oracle database access parameters need to be corrected in the `tokens_ora.properties` file to reflect the actual database connectivity.

c. Create a JDBC Resource for each application using the values from the `tokens_ora.properties` **file.**

- Resource JNDI Name: `jdbc/OracleFilesharingDB`. The value should match the *DB-JNDI-NAME* value in the `tokens_ora.properties` file.
- Resource Type: `javax.sql.DataSource`.
- Datasource Classname: `oracle.jdbc.pool.OracleDataSource`. The value should match *DB-DATASOURCE* value in the `tokens_ora.properties` file.
- User: `portalfs`. The value should match the `DB-USER` value in the `tokens_ora.properties` file.
- Password: `portalfs`. The value should match the `DB_PASSWORD` value in the `tokens_ora.properties` file.

- URL: `jdbc:oracle:thin:@oracle.acme.com:1521:portal`. The value should match the `DB-URL` value in the `tokens_ora.properties` file.

Note – On some web containers, you might need to define the JDBC Connection Pool connection pool prior to setting up the JDBC resource.

---

d. **Undeploy the existing portlets that use the Derby Database as the datastore.**

```
/opt/SUNWportal/bin/psadmin undeploy-portlet -u
uid=amadmin,ou=people,dc=acme,dc=com -f Password-File -p Portal ID -g portlet
```

e. **Deploy the newly configured portlet.**

```
cd Portal-Data-Dir/portals/Portal-ID/portletapps/Portlet_Application
```

```
/usr/sfw/bin/ant -D"portal.id=portal1" -Dapp.version=ora
```

CHAPTER 8

Basics of Portal Performance

Understanding the basic factors that affect Portal Server performance helps you to architect a well designed portal for your enterprise. For a well tuned portal, performance is determined by:

- Arrival rates (logins)
- User Activity (reloads)
- Desktop Configuration
- Backend systems

For an improperly tuned or sized portal, performance is determined by:

- CPU constraints
- Java Virtual Machine (VM) compilation and optimization
- Java VM Garbage Collection

The following sections cover the topics listed above and provide some guidelines to get the highest possible performance for Portal Server deployments. Performance is an algorithm analysis, based on many inputs or factors that determine portal server performance. Throughput is a principal consideration. It provides the data to consider for concurrent users that can be supported by your Portal server configuration and hardware. It is advisable to do a baseline load test to determine the characteristics of the architected system. A baseline test is one which tests the throughput with most common user/portal interaction Most of the data and performance testing tools (slamd) will display this data as RPS (Rounds per second). RPS and login per second convey the same meaning. Throughput is mainly CPU bound (user activity).

Performance Constraints

Portal Server performance can be CPU bound, memory bound or even bound by the capability of the garbage collector. The scenario where the portal is CPU bound is the easiest to detect and resolve. If CPU utilization at the point of peak load is greater than 75% the deployment will benefit from additional CPUs. Portal scales well to servers with 4 CPUs and it scales linearly when adding additional servers so long as the network itself is not an issue. Even if at the time of deployment a Portal Server installation is neither CPU nor memory bound a successful portal induces stickiness and the load increases over time. Consequently as with any web application it is paramount to monitor CPU, memory and all other logs consistently.

Performance Drivers

Some important general factors that affect portal performance and are not release dependent. These factors are the most important factors that should be considered by every customer and portal architect on every deployment.

- "Arrival Rates" on page 132
- "User Activity" on page 132
- "Desktop Channel Types" on page 133

Arrival Rates

Arrival rates are defined as the rate at which portal users (employees/customers) login over a period of time. Arrival rates with steep peaks can cause Portal Server to became CPU-bound even if at other times throughout the day CPU utilization is minimal. Consider the example where, employees and/or customers of an enterprise start connecting to the portal early in the morning, remain mostly idle for rest of day and either logout due to session time out or manually logout at end of business. This is the most common scenario for business to employee portals and for some outward-facing portals with high locality of their user base. The rate at which the users login in the morning or logout at the end of day and perform any interaction with the portal server, determines the throughput demanded from the portal server. The throughput test gives a reliable conclusion for this requirement.

User Activity

User activity is defined as the interaction that a portal users has with the portal server. It depends on the number of tabs, links, sites that a user has to go through to reach a specific channel or information.

User activity is also governed by the type of channels that a portal architect may decide to provide the users. For example, if the communications channels (default Mail and Calendar channels shipped with the Portal server software) are placed on a user's portal desktop, the user tends to demand updated content regularly. Such Portal sites with communication channels or connections to back-end systems face far shorter reload intervals . The affect of reloading the desktop is that it consumes CPU cycles and generates jvm heap garbage. Because portal is good at caching, sites that promote reloads can become memory bound before the CPU is stressed. It is recommended to use the JVM option for Concurrent garbage collector to alleviate this issue.

Desktop Channel Types

Configuration of the portal desktop can make a difference in portal server performance. While designing a portal desktop, the baseline data should be referenced as a starting point to guide through the desktop design process. A feature rich desktop is good to entice users to your portal, but you should consider the portal response time for an overall user experience. It is necessary to keep the desktop within a defined performance envelope. Types of channels used on the desktop make a huge difference. Some channels are considered cheap and others categorized as expensive . Cheap channels are the ones that when added do not affect performance and in contrast adding expensive channels will show some performance degradation. In general, the expensive channels are those that require a back-end service or are computationally expensive. It is generally advised to use connection pooling mechanisms for these. Adding a URL scrapper channel to the desktop generally shows almost no impact on throughput.

Portal Server Tuning

Tuning allows you to increase your Portal Server's performance.

- "Web Container Tuning" on page 133
- "Tuning Scripts" on page 134
- "Access Manager Tuning" on page 143
- "Directory Server Tuning" on page 143

Web Container Tuning

Of the tunings that can be applied to a Portal Server deployment, the JVM tunings of the web container are the most important. It is important to use the tuning scripts to apply that configuration and to understand the changes made by those scripts. To be able to run those scripts it is necessary to edit a configuration file.

The tunable parameters in this file are crucial for Portal Server performance. Review those parameters carefully, especially those that determine the amount of memory that will be given to the web container.

With significant improvement in the JVM scalability, the Portal application is also able to take advantage of the new JVM parameters resulting in increased performance. An optimal configuration of the web container calls for a 2 GB of Java heap. As far as binding to Portal Server instances to CPUs, we now have great scalability up to 4 CPUs on a single instance and to even more CPUs with multiple instances of PS.

Note – Each version of portal is certified for JVM tunings with the JVM version that ships with the product. Using a non-certified JVM is neither supported nor recommended. When deploying Portal Server in a production environment where all the component products are also deployed on the same server, it is recommended to use a machine with at least 4 GB or RAM.

Note – It is being assumed that the system is being exclusively used for Portal server. If Access Manager and Directory Server resources are being consumed by other applications, then you might need to consider the tunings appropriately.

Tuning Scripts

Access Manager performance utility `amtune` can be used to tune Directory Server, Access Manager and web container as well as operating system configurations. It is located in `/opt/SUNWam/bin/amtune/amtune` for a default install of Java Enterprise System software. Apart from the above, Portal product ships with a utility script called `perftune` (`/opt/SUNWportal/bin/`) that takes care of tuning the Directory server, Access Manager, web container, TCP/IP settings, Kernel settings and Portal Server configurations as well. In fact, `perftune` calls `amtune` to configure the former part of the tunings if they are deployed in the same web container. The administrator of Portal Server must understand the tunings recommended by the above utilities. These tunings should first be validated on a staging and quality assurance environment before being pushed to a production system.

AMTUNE

By default, `amtune` is configured to run in REVIEW mode. IN this mode, `amtune` will suggest tuning recommendations but will not make any changes to the deployment. This is a safe mode of running. This and other parameters are defined in a file called `/opt/SUNWam/bin/amtune/amtune-env`. The `amtune` script is a useful starting point to define the correct tuning parameters for the portal.

To invoke `amtune`, use the following command:

`amtune` *directory-server-admin-password web-server-admin-password*

The following is displayed:

```
Debug information log can be found in file:
/var/opt/SUNWam/debug/amtune-20061124-6300
############################################################
./amtune : 11/24/06 18:25:41
```

```
############################################################
Initializing...
-------------------------------------------------------------
Checking System Environment...
Checking User...
Checking Web Server JVM mode (32-bit or 64-bit) for
web server 7...
--------------------------------------------------------------
amtune Information...
--------------------------------------------------------------
amtune Mode      : REVIEW
OS               : true
Access Manager   : true
Directory        : true
Web Container    : true
WS Mode          : 32-bit
---------------------------------------------------------------
Detecting System Environment...
---------------------------------------------------------------
Number of CPUs in the system :  2
WS Acceptor Threads : 2
Memory Available (MB) :  2048
Memory to Use (MB) : 1536
There is enough memory.
----------------------------------------------------------------
Calculating Tuning Parameters...
----------------------------------------------------------------
Max heap size (MB) : 1344
Min Heap size (MB) : 1344
Max new size (MB) : 168
Cache Size (MB) : 448
SDK Cache Size (KB) : 298
Number of SDK Cache Entries : 38144
Session Cache Size (KB) : 149
Number of Session Cache Entries : 38144
Maximum Number of Java Threads : 672
Maximum Number of Thread Pool : 280
LDAP Auth Threads : 28
SM LDAP Threads : 28
Notification Threads : 14
Notification Queue Size : 38144
=================================================================
Access Manager Tuning Script
-----------------------------------------------------------------
Solaris Tuning Script
-----------------------------------------------------------------
Solaris Kernel Tuning...
```

```
File                 : /etc/system
Parameter tuning     :

1.   rlim_fd_max
Current Value        :   rlim_fd_max=
Recommended Value    :   rlim_fd_max=65536

2.   rlim_fd_cur
Current Value        :   rlim_fd_cur=
Recommended Value    :   rlim_fd_cur=65536


-----------------------------------------------------------------
Solaris TCP Tuning using ndd...

File                 : /etc/rc2.d/S71ndd_tcp
Parameter tuning     :

1.   /dev/tcp tcp_fin_wait_2_flush_interval
Current Value        :   /dev/tcp tcp_fin_wait_2_flush_interval 675000
Recommended Value    :   /dev/tcp tcp_fin_wait_2_flush_interval 67500

2.   /dev/tcp tcp_conn_req_max_q
Current Value        :   /dev/tcp tcp_conn_req_max_q 128
Recommended Value    :   /dev/tcp tcp_conn_req_max_q 8192

3.   /dev/tcp tcp_conn_req_max_q0
Current Value        :   /dev/tcp tcp_conn_req_max_q0 1024
Recommended Value    :   /dev/tcp tcp_conn_req_max_q0 8192

4.   /dev/tcp tcp_keepalive_interval
Current Value        :   /dev/tcp tcp_keepalive_interval 7200000
Recommended Value    :   /dev/tcp tcp_keepalive_interval 90000

5.  /dev/tcp tcp_smallest_anon_port
Current Value        :   /dev/tcp tcp_smallest_anon_port 32768
Recommended Value    :   /dev/tcp tcp_smallest_anon_port 1024

6.  /dev/tcp tcp_slow_start_initial
Current Value        :   /dev/tcp tcp_slow_start_initial 4
Recommended Value    :   /dev/tcp tcp_slow_start_initial 2

7.  /dev/tcp tcp_xmit_hiwat
Current Value        :   /dev/tcp tcp_xmit_hiwat 49152
Recommended Value    :   /dev/tcp tcp_xmit_hiwat 65536

8.  /dev/tcp tcp_recv_hiwat
Current Value        :   /dev/tcp tcp_recv_hiwat 49152
```

```
Recommended Value    :   /dev/tcp tcp_recv_hiwat 65536

9.  /dev/tcp tcp_ip_abort_cinterval
Current Value        :   /dev/tcp tcp_ip_abort_cinterval 180000
Recommended Value    :   /dev/tcp tcp_ip_abort_cinterval 10000

10.  /dev/tcp tcp_deferred_ack_interval
Current Value        :   /dev/tcp tcp_deferred_ack_interval 100
Recommended Value    :   /dev/tcp tcp_deferred_ack_interval 5

11.  /dev/tcp tcp_strong_iss
Current Value        :   /dev/tcp tcp_strong_iss 1
Recommended Value    :   /dev/tcp tcp_strong_iss 2


=====================================================================
Access Manager - Web Server Tuning Script
---------------------------------------------------------------------
Tuning Web Server Instance...

File                    : /var/opt/SUNWwbsvr7/https-xxxxxx.pstest.com/config/
server.xml (using wadm command line tool)
Parameter tuning     :

1.   Minimum Threads
Current Value        : min-threads=16
Recommended Value    : min-threads=10

2.   Maximum Threads
Current Value        : max-threads=128
Recommended Value    : max-threads=280

3.   Queue Size
Current Value        : queue-size=1024
Recommended Value    : queue-size=8192

4.   Native Stack Size
Current Value        : stack-size=131072
Recommended Value    : Use current value

5.   Acceptor Threads
Current Value        : acceptor-threads=1
Recommended Value    : acceptor-threads=2

6.   Statistic
Current Value        : enabled=true
Recommended Value    : enabled=false
```

```
7.   nativelibrarypathprefix
Current Value        : nativelibrarypathprefix=<No value set>
Recommended Value    : Append /usr/lib/lwp to nativelibrarypathprefix
(if Solaris 8)

8.   Max and Min Heap Size
Current Value        : Min Heap: -Xms512M Max Heap: -Xmx768M
Recommended Value    : -Xms1344M -Xmx1344M

9.   LogGC Output
Current Value        : <No value set>
Recommended Value    : -Xloggc:/var/opt/SUNWwbsvr7/https-xxxxxx.pstest.com/logs/gc.log

10.   JVM in Server mode
Current Value        : <No value set>
Recommended Value    : -server

11.   JVM Stack Size
Current Value        : -Xss128k
Recommended Value    : -Xss128k

12.  New Size
Current Value        : -XX:NewSize=168M
Recommended Value    : -XX:NewSize=168M

13.  Max New Size
Current Value        : -XX:MaxNewSize=168M
Recommended Value    : -XX:MaxNewSize=168M

14.  Disable Explicit GC
Current Value        : -XX:+DisableExplicitGC
Recommended Value    : -XX:+DisableExplicitGC

15.  Use Parallel GC
Current Value        : <No value set>
Recommended Value    : -XX:+UseParNewGC

16.  Print Class Histogram
Current Value        : <No value set>
Recommended Value    : -XX:+PrintClassHistogram

17.  Print GC Time Stamps
Current Value        : <No value set>
Recommended Value    : -XX:+PrintGCTimeStamps

18.  OverrideDefaultLibthread (if Solaris 8)
Current Value        : <No value set>
Recommended Value    : -XX:+OverrideDefaultLibthread
```

```
19.  Enable Concurrent Mark Sweep GC
Current Value        : <No value set>
Recommended Value    : -XX:+UseConcMarkSweepGC


=====================================================================
Access Manager - Directory Server Tuner Preparation Script
Preparing Directory Server Tuner...
---------------------------------------------------------------------
Determining Current Settings...
Creating Directory Server Tuner tar file: ./amtune-directory.tar
a amtune-directory 29K
a amtune-utils 45K

Directory Server Tuner tar file: ./amtune-directory.tar
Steps to tune directory server:
1. Copy the DS Tuner tar to the DS System
2. Untar the DS Tuner in a temporary location
3. Execute the following script in 'REVIEW' mode : amtune-directory
4. Review carefully the recommended tunings for DS
5. If you are sure of applying these changes to DS, modify the following
lines in amtune-directory
a. AMTUNE_MODE=
These parameters can also be modified or left unchange to use default values
b. AMTUNE_LOG_LEVEL=
c. AMTUNE_DEBUG_FILE_PREFIX=
d. DB_BACKUP_DIR_PREFIX=
Its highly recommended to run dsadm backup before running amtune-directory

=====================================================================
Access Manager - Access Manager Server Tuning Script
---------------------------------------------------------------------
Tuning /etc/opt/SUNWam/config/AMConfig.properties...

File                 : /etc/opt/SUNWam/config/AMConfig.properties
Parameter tuning     :

1.   com.iplanet.am.stats.interval
Current Value        : com.iplanet.am.stats.interval=60
Recommended Value    : com.iplanet.am.stats.interval=60

2.   com.iplanet.services.stats.state
Current Value        : com.iplanet.services.stats.state=file
Recommended Value    : com.iplanet.services.stats.state=file

3.   com.iplanet.services.debug.level
Current Value        : com.iplanet.services.debug.level=error
```

```
Recommended Value    : com.iplanet.services.debug.level=error

4.   com.iplanet.am.sdk.cache.maxSize
Current Value        : com.iplanet.am.sdk.cache.maxSize=10000
Recommended Value    : com.iplanet.am.sdk.cache.maxSize=38144

5.   com.iplanet.am.notification.threadpool.size
Current Value        : com.iplanet.am.notification.threadpool.size=10
Recommended Value    : com.iplanet.am.notification.threadpool.size=14

6.   com.iplanet.am.notification.threadpool.threshold
Current Value        : com.iplanet.am.notification.threadpool.threshold=100
Recommended Value    : com.iplanet.am.notification.threadpool.threshold=38144

7.   com.iplanet.am.session.maxSessions
Current Value        : com.iplanet.am.session.maxSessions=5000
Recommended Value    : com.iplanet.am.session.maxSessions=38144

8.   com.iplanet.am.session.httpSession.enabled
Current Value        : com.iplanet.am.session.httpSession.enabled=true
Recommended Value    : com.iplanet.am.session.httpSession.enabled=false

9.   com.iplanet.am.session.purgedelay
Current Value        : com.iplanet.am.session.purgedelay=60
Recommended Value    : com.iplanet.am.session.purgedelay=1

10.  com.iplanet.am.session.invalidsessionmaxtime
Current Value        : com.iplanet.am.session.invalidsessionmaxtime=10
Recommended Value    : com.iplanet.am.session.invalidsessionmaxtime=1


---------------------------------------------------------------------
Tuning /etc/opt/SUNWam/config/serverconfig.xml...

File                 : /etc/opt/SUNWam/config/serverconfig.xml

Recomended tuning parameters only. These paramters will not be tuned by the script.
You need to modify them manually in /etc/opt/SUNWam/config/serverconfig.xml.
The number should depend on number of Access Manager instances and the memory of
Directory Server.  Please refer to Access Manager Performance Tuning Guide.

1.   minConnPool
Current Value        : minConnPool=1
Recommended Value    : minConnPool=1

2.   maxConnPool
Current Value        : maxConnPool=10
Recommended Value    : maxConnPool=28
```

```
---------------------------------------------------------------------
Tuning LDAP Connection Pool in Global iPlanetAMAuthService...

Service              : iPlanetAMAuthService
SchemaType           : global

Recomended tuning parameters only. These paramters will not be tuned by the script.
If you want to tune these parameters, review data file /tmp/dsame-auth-core-tune.xml
and run it with amadmin command.  The number should depend on number of Access Manager
instances and the memory of Directory Server.  Please refer to Access Manager
Performance Tuning Guide.

1.   iplanet-am-auth-ldap-connection-pool-default-size
Recommended Value    : iplanet-am-auth-ldap-connection-pool-default-size=28:28

=====================================================================
Tuning Complete
#####################################################################
```

PERFTUNE

The `perftune` script runs the `amtune` script, but it also tunes the Portal Server. The following is an example of the `perftune` output:

```
Portal Tuning Script
---------------------------------------------------------------------
Tuning /var/opt/SUNWportal/portals/portal1/config/desktopconfig.properties...

File                 : /var/opt/SUNWportal/portals/portal1/config/
desktopconfig.properties
Parameter tuning     :

1.   callerPoolMinSize
Current Value        : callerPoolMinSize=0
Recommended Value    : callerPoolMinSize=128

2.   callerPoolMaxSize
Current Value        : callerPoolMaxSize=0
Recommended Value    : callerPoolMaxSize=256

3.   callerPoolPartitionSize
Current Value        : callerPoolPartitionSize=0
Recommended Value    : callerPoolPartitionSize=32

4.   templateScanInterval
Current Value        : templateScanInterval=30
```

```
Recommended Value    : templateScanInterval=3600

---------------------------------------------------------------------
Tuning /var/opt/SUNWportal/portals/portal1/config/PSLogConfig.properties...

File                 : /var/opt/SUNWportal/portals/portal1/config/
PSLogConfig.properties
Parameter tuning     :

1.   debug.com.sun.portal.level
Current Value        : debug.com.sun.portal.level=SEVERE
Recommended Value    : debug.com.sun.portal.level=FINE

=====================================================================
```

Thread Pools

The Java Virtual Machine (JVM) can support many threads of execution at once. To help performance, both Access Manger and Portal Server maintains one or more thread pools. Thread pools, allow you to limit the total number of threads assigned to a particular task. You can see an example of the tuning parameter `callerPool` in the output of `perftune` above. When a request is passed into the web container from a browser it will flow through several thread pools. A thread pool contains an array of `WorkerThread` objects. These objects are the individual threads that make up the pool. The `WorkerThread` objects will start and stop as work arrives for them. If there is more work than there are `WorkerThreads`, the work will backlog until WorkerThreads free up. Assigning an insufficient amount of threads in a thread pool can cause a bottleneck in the system which is hard to see. Assigning too many threads to a thread pool is also undesirable but normally is not critical.

RqThrottle — The *RqThrottle* parameter specifies the maximum number of simultaneous transactions a current Java Enterprise System web container can handle. The maximum number of simultaneous transactions is considered the thread limit.

Low-Memory Situations — If you need the web container to run in low-memory situations, reduce the thread limit to a bare minimum by lowering the value of *RqThrottle*. Also, you can reduce the maximum number of processes by lowering the value of the *MaxProcs* value. Typically this value will be `1` for the Java Enterprise System Web Server.

Under-Throttled Server — The server does not allow the number of active threads to exceed the thread limit value. If the number of simultaneous requests reaches that limit, the server stops servicing new connections until the old connections are freed up. By waiting for old connections to be freed, response time is increased. In Sun Java Enterprise System web containers, the server's default *RqThrottle* value is 128. If you want your server to process more requests concurrently, increase the *RqThrottle* value. The symptom of an under-throttled server is a server with a long response time. Making a request from a browser establishes a connection fairly quickly to the server, but on under-throttled servers it may take a long time before the response comes back to the client. The best way to tell if your server is being throttled is to see if the number of active sessions is close to, or equal to, the maximum number allowed by *RqThrottle*.

Access Manager Tuning

Portal server takes advantage of the Identity management and policy evaluation capabilities of the Access Manager and has been tightly integrated with the Access Manager solution. During the installation process of Portal server, the administrator is prompted for passwords and credentials of the Access Manager server.

Access Manager is also required to be tuned properly for Portal applications. The tuning script provided by AM, `amtune` takes care of most of the tuning required by the administrator for production systems and it is highly recommended to study these changes and understand their applicability. The `amtune` script output has been included earlier and it shows recommended changes in a format that makes it easy for a administrator to find and make those changes as required.

Directory Server Tuning

Portal Server and Access Manager both store their schema and user data in the Directory server. As the users of Portal Server or Access Manager customize their profiles, the desktop profile is saved as xml in the directory server. Is is important to tune the directory, index the most searched attributes for faster response, and tune the cache size for optimized performance.

Monitoring CPU Utilization

The `mpstat` utility on Solaris can be used to monitor CPU utilization, especially with multi-threaded applications running on multiprocessor machines, which is a typical configuration for enterprise solutions.

The `mpstat` command reports processor statistics in tabular form. Each row of the table represents the activity of one processor. The first table summarizes all activity since boot. Each subsequent table summarizes activity for the preceding interval. All values are rates listed as events per second unless otherwise noted.

Use `mpstat` with an argument between 5 seconds to 10 seconds. An interval that is smaller than 5 or 10 seconds might be more difficult to analyze. A larger interval might provide a means of smoothing the data by removing spikes that could mislead the result.

Input

```
mpstat 10
```

Output

```
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
0 1 0 5529 442 302 419 166 12 196 0 775 95 5 0
0 1 1 0 220 237 100 383 161 41 95 0 450 96 4 0
0 4 0 0 27 192 100 178 94 38 44 0 100 99 1 0 0
```

What to Look For

The cpu utilization is the inverse of idle time. ie 0% idl means 100% cpu utilization.

Monitoring Memory Utilization

The `vmstat` utility on UNIX can be used to monitor memory utilization. The most important criteria after having sufficient physical memory in the machine is to have sufficient swap space to sustain the virtual memory requirements of the portal/access manager web containers, directory process's and other associated process's.

The following example shows the input and output.

Input

```
% vmstat 5
```

Output

```
kthr   memory          page             disk      faults        cpu
      r b w swap  free re mf pi p fr de sr s0 s1 s2 s3  in  sy  cs us sy id
      0 0 0 11456 4120 1  41 19 1  3  0  2  0  4  0  0  48 112 130  4 14 82
      0 0 1 10132 4280 0   4 44 0  0  0  0  0 23  0  0 211 230 144  3 35 62
```

◆ ◆ ◆ APPENDIX A

Understanding Portal Server and Application Servers

Sun Java System Portal Server deployments require a web container. The web container, which manages the components of your portal applications, is typically the Sun Java System Application Server or Sun Java System Web Server.

- "Application Server Support in Portal Server" on page 145
- "Portal Server on an Application Server Cluster" on page 146

Application Server Support in Portal Server

Portal Server provides supports for three application servers.application servers to be used as the web application container, in addition to the Java™ Web Server software:

- Sun Java SystemApplication Server 8.2
- BEA WebLogic Server™ Server 8.1 SP4 and 8.1 SP5
- IBM WebSphere® Application Server 5.1.1.6

Running Portal Server on an application server enables you to:

- Decouple the portal platform from the application server platform, allowing you to choose the best combination of Portal Server and application server for your organization
- Call Enterprise JavaBeans™ architecture and other J2EE™ technologies that run in the application server container
- Use application server clustering, which provides scalability and high availability
- Use session failover in clustering (currently available on BEA WebLogic Server andApplication Server Enterprise Edition)
- Provide portlet session failover

Portal Server on an Application Server Cluster

This section describes how Application Server Enterprise Edition software, BEA WebLogic Server, and IBM WebSphere Application Server manage *application server clustering*. Application server clustering is a loosely coupled group of application servers that collaborate to provide shared access to the services that each server hosts. The cluster aims to balance resource requests, high availability of resources, and failover of application logic to provide scalability. Portal Server and Access Manager are not pure web applications. Instead, these applications are composed of local files residing on a machine and 17 default web applications. Many of the default web applications are portlets. The number of applications by default is:

- portal
- amserver
- ampassword
- amcommon
- amconsole
- search1
- psconsole
- communityportlets
- filesharing
- portletsamples
- ipc1
- ipc2
- rssportlet
- guessnumber
- wsssoportlet
- surveys
- wiki

The Java Enterprise System installs and configures the local files, configures the local application server, then deploys the WAR files on the local web application container. The WAR files themselves are not self-contained. The WAR files depend on the local files and directories on the machine to provide their service.

An application server cluster is a logical entity that groups many application server instances, potentially hosted on different machines. Pure web applications are deployed on a cluster using application server specific deployment tools. Once deployed on the cluster, the web applications are deployed to all the server instances that the cluster is made of, and managed in a central way.

Because of Portal Server's dual nature, as a local application as well as a web application, install Portal Server on an application server using the following steps:

▼ To Install Portal Server on an Application Server

1. **Install Portal Server on all machines using the same configuration settings.**

2. **Deploy the web applications (portal, amserver, and psconsole . . .) to the cluster.**

   The following sections explain what it means to enable Portal Server to run on an application server cluster.

Application Server Enterprise Edition

The Sun Java System Application Server Enterprise Edition 8 provides a robust J2EE platform for the development, deployment, and management of enterprise applications. Key features include transaction management, performance, scalability, security, and integration. The Application Server supports services from Web publishing to enterprise-scale transaction processing.

The Application Server is available in the Platform and Enterprise editions. The Platform edition is free and is intended for software development and department-level production environments. Designed for mission-critical services and large-scale production environments, the Enterprise edition supports horizontal scalability and service continuity via a load balancer and cluster management. The Enterprise edition also supports session continuity via the Highly Available Database (HADB). See the following Application Server Enterprise Edition documentation for more information:

http://docs.sun.com/db/coll/1310 ()

BEA WebLogic Server

The BEA WebLogic Server product uses the following definitions:

- **Domain**. An interrelated set of WebLogic Server resources managed as a unit. A domain includes one or more WebLogic Servers, and might include WebLogic Server clusters.
- **Administration Server**. A WebLogic Server running the Administration Service. The Administration Service provides the central point of control for configuring and monitoring the entire domain. The Administration Server must be running to perform any management operation on that domain.
- **Managed Server**. In a domain with multiple WebLogic Servers, only one server is the Administration Server; the other servers are called Managed Servers. Each WebLogic Managed Server obtains its configuration at startup from the Administration Server.

See the following documentation for more information:

`http://edocs.beasys.com/wls/docs81/cluster/overview.html#1000572`

You start the Administration Server with the following command:

install_dir`/config/domain_name/startWeblogic.sh`

The local server takes its configuration from the *install_dir*`/config/domain_name/config.xml` file.

To start a Managed Server, use the following command:

install_dir`/config/domain_name/startManagedWebLogic.sh` *servername admin_server_url*

Instead of taking its configuration from the *install_dir*`/config/domain_name/config.xml` local file, the Managed Server takes it from the Administration Server, using HTTP.

Note – The default configuration supported for installing Portal Server on BEA WebLogic Server is a single server that is also the Administration Server for the domain.

A BEA cluster is a set of managed servers in the same domain, that are declared in the WebLogic console as a cluster. When deploying a web application, you use the name of the cluster, not the name of the individual servers. After the deployment, the web application is identically deployed to all machines in the cluster.

Session failover in BEA is described in the following document:

`http://edocs.beasys.com/wls/docs81/cluster/failover.html#1022034`

Using in-memory replication for HTTP session states requires the following prerequisites:

- Portal Server supports the use of WebLogic Server clusters with in-memory session replication. See the BEA documentation for instructions to set up these clusters. The *Sun Java Enterprise System 2005Q5 Installation Guide* documents the load balancer configuration for such a cluster using the HttpClusterServlet that ships with BEA. You can also set up other load balancing hardware and software documented by BEA in the same way.
- Session data must be serializable.
- Use the `setAttribute` to change the session state.

To install a BEA cluster, your BEA license for each machine participating in the cluster must be a special BEA cluster license. See the BEA documentation for the procedure to get the license and set up a BEA cluster with HttpClusterServlet.

IBM WebSphere Application Server

The IBM WebSphere Application Server product uses the following definitions:

- **Deployment Manager** The deployment manager is the central administrative manager. It does not install the base WebSphere Application Server product on other machines. Only one system hosts the deployment manager.
- **Cell** A cell is made up of nodes (servers) containing node agents, which are managed by a deployment manager. As the deployment manager federates base WebSphere Application Server nodes, it expands the cell that it manages.
- **Clusters** A cluster is a group of servers that can be used together for workload balancing by creating additional, nearly identical copies of an application server configuration. This group is the equivalent of BEA's cluster. Clusters are identical multiple Application Server copies. See the IBM WebSphere Application Server documentation for more information: `http://www-306.ibm.com/software/webservers/appserv/was/library/library51.html`

  WebSphere Application Server Network Deployment edition addresses application servers that run in multiple-server production scenarios. It provides centralized administration, as well as basic clustering and caching support.

WebSphere Advanced Server Network Deployment provides a more robust approach to clustering because it includes a database. In Advanced Server, all servers use the database for the configuration information. You can use the WebSphere administration console, a Swing Java application, or the command-line utilities `XMLConfig` and `wscpthen` to manage the servers.

◆ ◆ ◆ APPENDIX B

Troubleshooting Your Deployment

This appendix describes how to troubleshoot the Sun Java^{TM} System Portal Server software and the Sun Java System Portal Server Secure Remote Access software.

This appendix contains the following sections:

- "Troubleshooting Portal Server" on page 151
- "Troubleshooting Secure Remote Access" on page 157

Troubleshooting Portal Server

This sections contains troubleshooting information for Sun Java System Portal Server.

Troubleshooting the Portal Server Installation

Examining the Portal Server log files helps you to identify errors that might have occurred during the installation phase of your Portal Server deployment. The following sections discuss the log file locations and information:

- "Installation Log Files" on page 151
- "Portal Configuration Log Files" on page 152
- "Portal Admin and MBeans Log Files" on page 153

Installation Log Files

The Java Enterprise System installer log files are located in:

`/var/sadm/install/logs/`. These log files contain information about the portal packages installed and the basic configuration status. The files are:

- Java_Enterprise_System_5_Summary_Report_install.*timestamp*
- Java_Enterprise_System_5_install.B.*timestamp*

- Java_Enterprise_System_5_install.A.*timestamp*
- JavaES_Install_log.*timestamp*

▼ To Check for Install Errors

- **Use the following command to view install errors:**

```
grep -i failed /var/sadm/install/logs/*
```

Example B–1 Java Enterprise System Install Logs

```
 Java_Enterprise_System_5_Summary_Report_install.timestamp
***********extract************
Sun Java(TM) System Access Manager 7.1 : Installed, Configured
Service Registry 3.1 : Installed, Configure After Install
Sun Java(TM) System Portal Server 7.1 : Installed, Configuration Failed
Sun Java(TM) System Portal Server Secure Remote Access 7.1 : Installed, Configuration
Failed
*************************

JavaES_Install_log.timestamp
***********extract************
Creating Portals
Successfully created Portal: portal1|#]
[#|2006-11-21T14:22:15+00:00|WARNING|JavaES|JavaESInstall|
_versionID=1.0;_threadID=13;_SourceJavaFile=EntsysConfigurator;
_SourceMethodName=executeCmd(cmdArray);_JavaESMessageID=;_JavaESResourceBundle=|
Configuration Failed : com.sun.portal.fabric.tasks.ConfigurationException:
javax.management.MBeanException:
Exception thrown in operation createAndFinalizeInstance|#]
****************************
```

Portal Configuration Log Files

The Portal Server configuration log files are located in:

```
/var/opt/SUNWportal/logs/config/portal.fabric.0.0.log
```

The following example show a portion of the Portal Configuration log file.

```
***********extract************
[#|2006-11-21T14:22:14.109+0000|INFO|SJS Portal Server|
debug.com.sun.portal.fabric.config|
ThreadID=10; ClassName=com.sun.portal.fabric.config.PortalConfigurator;
MethodName=configurePortal; |PSFB_CSPFC0032:Creating Portal Instances|#]

[#|2006-11-21T14:22:14.509+0000|SEVERE|SJS Portal Server|
debug.com.sun.portal.fabric.config|
ThreadID=10; ClassName=com.sun.portal.fabric.config.PortalConfigurator;
MethodName=createPortalInstances; |PSFB_CSPFC0041:Failed invoking mbean action :
create instance.
javax.management.MBeanException: Exception thrown in operation
createAndFinalizeInstance
        at com.sun.jmx.mbeanserver.StandardMetaDataImpl.invoke
(StandardMetaDataImpl.java:435)
Caused by: com.sun.portal.admin.common.PSMBeanException:
com.sun.portal.fabric.tasks.ValidationException: Could not fetch
the value of Platform. Please check wadm username and password
****************************
```

Portal Admin and MBeans Log Files

The Portal admin and MBeans logs are available at:

```
/var/opt/SUNWportal/logs/admin/portal.0.0.log
```

Checking the Configured Portals

You can find which portals are configured using the psadmin command. The following command shows which portals are configured.

Input

```
PATH=/opt/SUNWportal/bin:/usr/jdk/entsys-j2se/bin:${PATH};export PATH
```

```
psadmin list-portals -u amadmin -f /tmp/passwd
```

Output

```
Domain: defaultDomain
============================================
Portal: portal1
============================================
```

Checking Unix Processes

For the portal to be functioning properly, check that the following `root`-owned processes are running. Use the `ps` command to see this output.

Sun Java System Directory Server:

```
ps -ef | grep dsee
```

```
ns-slap -D /var/opt/SUNWdsee/dsins 1 -i /var/opt/SUNWdsee/dsins1/logs/pid
```

Sun Java System Access Manager:

Check for web container processes (webserver)

```
ps -ef | grep webserv
```

```
webservd -d /var/opt/SUNWwbsvr7/https-xxxx.pstest.com/config -r /opt/SUNWwb
```

Sun Java System Portal Server:

Check for web container processes (Application Server)

```
ps -ef | grep appserver
```

```
/opt/SUNWappserver/appserver/lib/appservDAS domain1
```

Cacao Server:

Verify that the cacao process is running

```
/usr/ucb/ps -auxww | grep cacao
```

```
/usr/jdk/jdk1.5.0_09/bin/java -Xms 128M -Xmx256M ...-classpath
.../usr/lib/cacao/lib/cacao_cacao.jar ...
Dcacao.config.dir=/etc/cacao/instances/default ...
```

Derby database:

```
ps -ef | grep derby
```

Admin Web Server (optional, but usually running):

```
./uxwdog -d web-container-install-root/SUNWam/servers/https-admserv/config
ns-httpd -d web-container-install-root/SUNWam/servers/https-admserv/config
```

Runtime Log Files

Examine the following log files for errors.

Access Manager logs are located in:

```
/var/opt/SUNWam/debug
```

Runtime Portal Server logs:

Runtime Portal instance logs are available at:

```
/var/opt/SUNWportal/portals/portal-server-ID/logs
```

Restarting Processes

The following tasks describe how you restart some of the component processes.

▼ To Restart the Web Server

1. **Use the following command to go to the Web Server instance directory.**

   `cd /var/opt/SuNWwbsvr7/https-`*instance-name*`.pstest.com/bin`.

2. **Stop the Web Server.**

   `./stopserv`.

3. **Start the Web Server.**

   `./startserv`.

   The following message will be displayed when the portal module is loaded.

   info: WEB0100: Loading web module in virtual server [xxxx.pstest.com] at [/portal]

▼ To Restart Cacao

1. **Use the following command to go to the cacao directory:**

   `cd /usr/lib/cacao/bin`.

2. **Stop the cacao server.**

   `./cacaoadm stop`.

3. **Start the cacao server.**

   `./cacaoadm start`.

▼ To Restart the Directory Server

● **Use the following command to restart the Directory Server.**

Directory-Server-install-dir`/SUNWdsee/ds6/bin/dsame start /var/optSUNWdsee/dsins1`

Working with the Display Profile

If you need to troubleshoot the XML contents of your portal's display profile, extract the contents to a file for examination. At some point in the troubleshooting process, it might be useful to reload the display profile.

▼ To Extract the Display Profile

1. **Login as administrator.**
2. **Use the `psadmin` command to extract the display profile. For example:**

   ```
   psadmin list-display-profile -u uid=amadmin,ou=People,dc=pstest,dc=com -f
   /tmp/passwd -p portal1 -o /tmp/global_dp.xml -g
   ```

   This example puts the contents of the display profile into the `/tmp/global_dp.xml` file.

▼ To Reload the Display Profile

1. **Login as administrator.**
2. **Use the `psadmin` command to reload the display profile. For example:**

   ```
   psadmin modify-display-profile -u uid=amadmin,ou=People, dc=pstest,dc=com -f
   /tmp/passwd -d uid=user001,ou=People,o=DeveloperSample,dc=pstest,dc=com -p
   portal1 /tmp/updated_display.xml
   ```

   This example reloads the contents of the display profile from the `/tmp/updated_displayxml` file.

Configuring an HTTP Proxy

If the Portal Server software is installed on a host that cannot directly access certain portions of the Internet or your intranet, you can receive errors. For example, when using the `SampleSimpleWebService` provider, you might see the following error when the proxy has not been configured:

```
HTTP transport error: java.net.ConnectException: Connection refused
```

▼ To Configure Usage of an HTTP Proxy for a Portal Server Instance

The following procedure describes how to configure an HTTP Proxy for a Portal Server instance on Sun Java System Web Server.

1. **Change directories to the web container data directory containing the configuration for the instance.**

   `cd` *webcontainer-install-root*`/SUNWwbsvr7/https-`*instance-name*`/config`

   For example: `/var/opt/SUNWwbsvr7/https-siroe.com/config`

2 **Edit the `server.xml` file within this directory and add the following lines:**

```
<jvm-options>-Dhttp.proxyHost=proxy-host</jvm-options>

<jvm-options>-Dhttp.proxyPort=proxy-port</jvm-options>

<jvm-options>-Dhttp.nonProxyHosts=portal-host</jvm-options>
```

Where *proxy-host* is the fully-qualified domain name of the proxy host, *proxy-port* is the port on which the proxy is run, and *portal-host* is the fully qualified domain name of the portal host.

3 **Restart the webcontainer for the changes to take affect.**

Troubleshooting Secure Remote Access

This section describes how to capture information that Portal Server support personnel need to troubleshoot problems in your deployment.

▼ To Check Secure Remote Access Status

● **Use the following command to check the status of Secure Remote Access:**

```
get-sra-status -u amadmin -f /tmp/pwdfile
```

The following response is returned:

```
on
```

▼ To Enable Secure Remote Access After Installation

● **Use the following `psadmin` cli command to enable the Secure Remote Access core after installation:**

```
portal-server7.1-base/bin/psadmin switch-sra-status -u amadmin -f /tmp/pwdfile
```

The following response is returned:

```
on
```

▼ To List the Secure Remote Access Instance

● **Use the following `psadmin` command to list the Secure Remote Access Instance:**

```
portal-server7.1-base/bin/psadmin list-sra-instances -u amadmin -f /tmp/pwdfile -t
gateway.
```

The following is displayed:

default:*hostname*.pstest.com|*ip-address*

▼ To Start Secure Remote Access Instance

● **Use the following `psadmin` cli command to start the Secure Remote Access instance:**

portal-server7.1-base/bin/psadmin start-sra-instance -u amadmin -f /tmp/passwd -N default -t gateway

The following is displayed:

```
For gateway-profile default, Secure Remote Access is not provisioned for any
portal. Please run psadmin provision-sra for gateway-profile default or modify
enableSRAForPortal.xml file for gateway-profile default and upload using amadmin
before attempting to start the sra-instance.
```

▼ To Provision Secure Remote Access Instance

● **Use the following `psadmin` cli command to provision a Secure Remote Access instance:**

```
psadmin provision-sra -u amadmin -f /tmp/passwd -p portal1 --gateway-profile
default --enable
```

▼ To setup Non-authenticated URL List for Secure Remote Access

● **Edit the non authenticate URL list:**

Edit a copy of the file, `portalserver7.1_base/export/request/enableSRAforPortal.xml` with correct values.

Note – Edit the `enableSRAforPortal.xml` file to enable the unathenticated access to the portal desktop and to apply the default rewriter rules.

- To enable the unauthenticated access to the Portal desktop, edit `sunPortalGatewayNonAuthenticatedURLPath`.
- To apply the default rewriter rules edit, `sunPortalGatewayDomainsAndRulesets`.

  Use the following command:

  ```
  amadmin -u uid=amAdmin,ou=People,dc=pstest,dc=com -w password --data
  /opt/SUNWportal/export/request/enableSRAforPortal.xml --verbose --continue
  ```

Debugging the Gateway

▼ To Check the Gateway Process

● **Use the following command to see if the gateway process is running:**

```
/usr/ucb/ps -auxww | grep SRAP
```

The following response is displayed:

```
/usr/jdk/entsys-j2se/bin/java -Dgateway.profilename=default ...
-Dgateway.notification.url=notification
-Dgateway.keybase=/etc/opt/SUNWportal/cert/default
-Dgateway.pass=/etc/opt/SUNWportal/cert/default/.jsspass
-Dgateway.nickname=/etc/opt/SUNWportal/cert/default/.nickname
-DLOG_COMPATMODE=Off
-Djava.util.logging.config.file=/opt/SUNWam/lib/LogConfig.properties
-Dcom.sun.portal.log.config.file=/etc/opt/SUNWportal/platform.conf.default
-Dconf.suffix=default -Dserver.name=default
-DSRAP_CONFIG_DIR=/etc/opt/SUNWportal com.sun.portal.netlet.eproxy.EProxy &
```

▼ To Use Debugging

To turn debugging on or off, you set the level of debugging or set it to off. The following steps describe what to do.

1. **Log in as root to the Gateway machine and edit the following file:**

   ```
   /etc/opt/SUNWportal/platform.conf.default
   ```

2. **You can enable to following debug options:**

   - `debug.com.sun.portal.rewriter.original.level`
   - `debug.com.sun.portal.level`

- `debug.com.sun.portal.rewriter.rulesetinfo.level`
- `debug.com.sun.portal.rewriter.uriinfo.level`

```
com.iplanet.services.debug.level=FINEST
```

The debug levels are:

- `OFF` – No debug messages are logged.
- `Severe` – Only serious errors are logged in the debug file. Rewriter usually stops functioning when such errors occur.
- `WARNING` – Is a message level indicating a potential problem.
- `FINEST` - Indicates a highly detailed tracing message.

Secure Remote Access Log Files

Examine the following log files for errors.

```
/var/opt/SUNWportal/logs/sra/default/*log
```

APPENDIX C

Portal Deployment Worksheets

This appendix provides worksheets to help with the portal deployment process.

This appendix contains the following sections:

- "Portal Assessment Worksheets" on page 161
- "Portal Design Task List" on page 164

Portal Assessment Worksheets

Use these worksheets to learn more about your organization's business needs and potential areas of concern around deploying portals.

TABLE C–1 General Questions

Identify the business reasons why you want a portal (check and elaborate on all that apply): ■ Reducing procurement cost ■ Reducing the cost of sharing information with customers, suppliers, or partners ■ Reducing the time to deploy new business services ■ Eliminating the cost of maintaining many point solutions ■ Expanding the reach of the customer base for your services ■ Securing the access to your data and services ■ Making it easier for your customers to do business with you over the Internet ■ Reducing the cost and time for integrating business services with suppliers and partners ■ Complying with governmental regulations ■ Personalizing the user experience ■ Needing to gather business intelligence on the usage of services ■ Providing a way for users to set up communities and to interact with other community members
How many portals does your organization already have? If you have more than one, do you have a need to reduce the number? Integrate? Federate? What types are they (business-to-employee, business-to-consumer, business-to-business, ISP)? Do you have departmental portals?
What is the extent of your Web presence? How many web sites do you have?
List the top ten application services of value to you, that you would like to expose to your partners? Suppliers? Customers? Employees?
Who is the target community for your portal?

TABLE C–2 Organizational Questions

Who are the stakeholders of this portal?
Who are the business owners (department, organization, or an individual) within your organization who would expose the content or application service that they own by using the portal?
Would an application service exposed by using the portal be made up of smaller business applications managed by an inter-departmental business process?
Who would "own" this portal (the infrastructure)?
Who would own the content?
How do you plan to recruit additional business owners within your organization to contribute their content or applications for your portal?

TABLE C–2 Organizational Questions *(Continued)*

What project management, architect, and technical implementation resources do you have available to help develop this portal?
Who sets the policies for web site characteristics such as look and feel and presentation?

TABLE C–3 Business Service-level Expectations Questions

Are your development projects consistent? Do you manage their risk?
How does your development team work with your test, deployment, and operations groups?
How many different platforms does your organization currently support?
How secure is your information? How consistent is the security?
Are these challenges getting better, or getting worse?

TABLE C–4 Content Management Questions

Do you have a content or document management system?
Do you have any defined workflow to manage the development and publication of content?
Do you have a taxonomy defined?
How well is your information tagged and categorized?
How is your enterprise content developed, managed, tracked, and published?
Do you have a need for syndicated content on your portal? If so, what?
What proportion of your content is dynamic versus static?

TABLE C–5 User Management and Security Questions

How would you segment, categorize, and relate (hierarchically) your user community?
What are your current and future security policies?
Do various departments own or maintain their private view of the customer?
Do you have an enterprise directory?

TABLE C–6 Business Intelligence Questions

Do you have a need to gather, store, analyze, and provide information for enterprise decision-making?
Do you already employ any data analysis or Online Analytical Processing (OLAP) tools?
At what level(s) do you need to collect business intelligence (enterprise-wide, division, department, project, onetime event)?

TABLE C–7 Architecture Questions

Do you already have an existing architecture strategy? Are there organizational issues that are hindering a successful implementation of a new IT architecture? Do you have the capabilities to implement a new architecture solution? What technologies do you currently use? Do you have the staff to implement a new architecture solution?
For the top ten services that you would like deployed by using a portal, what platform and architecture do you need to support?
How do these services authenticate users and manage access control
How do you programmatically gain access to these services?
What is your current and future messaging (email) and collaboration architecture? What is your current and future enterprise directory architecture?
What technologies are used for application integration?
What is the size of the target user community?
How many concurrent users?
What is the range of portal usage?
What is the geographical distribution of your user base?
Do you currently have or have a future need for non-Web access (Wireless, Voice/IVR)
Would your customer base require internationalization of content and services?
What server platform technologies do you use?
What development environments, tools do you use? What development methodologies do you employ?

Portal Design Task List

Table C–8 lists the major portal deployment phases and design tasks. Use this task list to help develop your portal project plan.

Though these tasks will vary depending on your organization and the scale of each deployment, the worksheet represents the most common phases and tasks encountered.

This table consists of two columns. The first column presents the major tasks. The second column presents the subtasks for each major task.

TABLE C–8 Design Task List

Major Phases and Tasks	Subtasks
1. Project Start and Coordination	
Project Planning	■ Perform general project management
Project Plan Review	■ Review pre-implementation ■ Review business requirements ■ Review technical requirements ■ Review architectural documents ■ Review hardware and infrastructure
Coordinate Resources	■ Identify skills required ■ Identify resources ■ Schedule resources ■ Assemble project team members ■ Review work plan with project team members
Define Requirements	■ Collect business requirements ■ Summarize requirements ■ Confirm functional requirements ■ Collect technical requirements ■ Summarize technical requirements ■ Confirm technical requirements ■ Prepare combined requirements document ■ Deliver requirements
2. Design	
Develop Solution Architecture	■ Design software architecture ■ Design server topology ■ Document architecture
Develop Portal Integration	■ Understand system integration approach ■ Define container and channel layout ■ Define content aggregation ■ Define SSO approach ■ Develop custom Netlet and authentication modules
User Interface Design	■ Prepare or modify user interface design ■ Develop or update screen specifications ■ Review and approve user interface model

TABLE C–8 Design Task List (Continued)

Major Phases and Tasks	Subtasks
Directory Design	Data design ■ Schema design ■ Directory tree design ■ Topology design ■ Replication design ■ Security design ■ Tuning and optimizations ■ Operations and decisions
3. Develop and Integrate	
Install Software for Testing Environments	■ Install supporting Directory Server ■ Install appropriate Web Container (Application Server or Web Server) ■ Install Access Manager ■ Install Sun Java System Portal Server software and optionally Secure Remote Access software (install appropriate supporting software) ■ Install other software ■ Configure server software ■ Test server software components ■ Document test findings
Install Server Software for Development Environment	■ Install supporting Directory Server ■ Install appropriate Web Container (Application Server or Web Server) ■ Install Access Manager ■ Install Portal Server and optionally Portal Server Secure Remote Access ■ Install other software ■ Configure server software ■ Test server software components ■ Document test findings
Software Configuration	■ Apply specific software configuration requirements ■ Create product configuration matrix

TABLE C–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Portal Server, Application Server, and Other Software Modifications	■ Review your organization's requirements and expectations ■ Establish modifications for software ■ Establish methods for software modifications ■ Create software modification plan ■ Design software modifications ■ Establish software modification teams ■ Create modifications ■ Test modifications ■ Obtain appropriate stakeholder and organizational review and approval of modifications
LDAP Directory Setup	■ Confer with stakeholders to establish proper schema ■ Establish modifications for software ■ Establish methods for software modifications ■ Create software modification plan ■ Design software modifications ■ Establish software modification teams ■ Create schema ■ Set up LDAP ■ Receive and verify data ■ Modify mapping as required for LDAP ■ Establish data update methods ■ Test directory ■ Create client user documentation for update methods
Legacy Software Integration (such as PeopleSoft, SAP)	■ Perform integration ■ Prepare package integration test plan ■ Perform integration test ■ Produce package integration test results
Reporting	■ Establish reporting requirements for organization ■ Create reporting plan ■ Establish reporting team ■ Design reports ■ Create reports ■ Test reports ■ Review reports with customer ■ Provide information and training on report tool
Test	■ Establish test plan

TABLE C–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Plan User Acceptance Test	■ Identify user acceptance test manager ■ Develop user acceptance test strategy and procedures ■ Review strategy and procedures with customer ■ Obtain approval for strategy and procedures ■ Develop user acceptance test roles and responsibilities ■ Obtain integration test scenarios ■ Review test conditions and acceptance criteria and revise ■ Develop user acceptance test schedule ■ Prepare acceptance test log and update with scenario test assignments
Conduct User Acceptance Test	■ Execute user acceptance test ■ Identify and document user acceptance test discrepancies ■ Resolve user acceptance test discrepancies ■ Re-execute user acceptance tests and track user acceptance test progress ■ Catalog and prioritize known limitations and process improvement opportunities identified during testing ■ Review test results with quality assurance advisors, summarize and communicate results to stakeholders ■ Obtain acceptance test approval from stakeholders

TABLE C–8 Design Task List (Continued)

Major Phases and Tasks	Subtasks
Conduct Integration and System Test	■ Ensure establishment of integration test environment ■ Identify test team and assign test scenario ownership ■ Train team on integration test procedures, roles, and responsibilities ■ Review and revise integration test execution schedule, as required ■ Execute integration test ■ Identify and document integration test discrepancies ■ Resolve integration test discrepancies and document ■ Identify required modifications (such as configuration enhancements, interfaces, reports) ■ Re-execute integration tests ■ Update as required ■ Track test progress ■ Obtain test approval ■ Summarize and communicate results to stakeholders
4. Deployment Production	
Confirm Approach	■ Review with stakeholders and establish implementation locations and configurations ■ Develop implementation approach ■ Repeat appropriate tasks from development hardware and software installation
Review and Update Deployment	■ Review existing documentation of results of tests ■ Validate scope, objectives, and critical success factors ■ Update deployment approach ■ Review and approve deployment
Implement Deployment	■ Review and reconcile system operations ■ Review organization and system procedures ■ Promote to production ■ Update current operations ■ Revise system release and deployment materials ■ Provide transition support

TABLE C–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Training	■ Confirm organization commitment and expectations ■ Establish training requirements for all personnel ■ Establish training schedules ■ Establish training staff ■ Prepare materials for training ■ Train administrators ■ Train maintenance providers ■ Capture training feedback ■ Incorporate feedback for training improvement
Document Portal	■ Create a "run book" for system administrators

APPENDIX D

Understanding How Files and Directories Are Installed

This appendix describes the Sun Java™ System Portal Server directory structure and properties files used to store configuration and operational data.

Directories Installed for Portal Server

Table D–1 shows the Solaris Sparc platform-specific directory structures that are installed for Portal Server.

TABLE D–1 Portal Server Directories

Description	Location
Default installation directory	*portal-server-install-root/SUNWportal*
Default installation directory for configuration information	`/etc/opt/SUNWportal`
Default installation directory for SDK	*portal-server-install-root*`/SUNWportal/sdk`
Temporary files	`/usr/tmp`

Log files	`/var/opt/SUNWam/log`
	`/var/opt/SUNWportal/portals/`*portal-instance*`/logs`
	The default instance directory is `portal1`

Search Engine logging, configuration, and data directories	`/var/opt/SUNWportal/searchservers/`*search-server-instance* The default search server instance name is `search1`

TABLE D–1 Portal Server Directories *(Continued)*

Description	Location
Container and channel display profile	*portal-server-install-root*/`SUNWportal/par-src/default-portal/dp/global-desktop.xml`
HTML template files	`/var/opt/SUNWportal/portals/`*portal-instance*`/desktop/default/`*provider-name*`/html` The default portal instance name is `portal1`
JSP template files	`/var/opt/SUNWportal/portals/`*portal-instance*`/desktop/default/*jsp` The default portal instance name is `portal1`
Command-line utilities	*portal-server-install-root*/`SUNWportal/bin/`
Tag library definitions	`/var/opt/SUNWportal/portals/portal1/desktop/default/tld/*tld`
Display profile DTD	`/etc/opt/SUNWportal/dtd/psdp.dtd`
Java properties files	`/var/opt/SUNWportal/portals/portal1/desktop/classes/`

Directories Installed for Secure Remote Access

This section describes the Sun Java System Secure Remote Access directory structure and configuration files used to store configuration and operational data.

Table D–2 shows the platform-specific directory structures that are installed for Secure Remote Access.

TABLE D–2 Portal Server Secure Remote Access Directories

Description	Location
Default installation directory	*portal-server-install-root/*
Default installation directory for configuration information	`/etc/opt/SUNWportal`
Log files	`/var/opt/SUNWportal/logs`

Log File Locations

Log files provide information about installation status or error reporting. Table D–3contains a list of log files and locations that you can use to troubleshoot and view install information.

TABLE D–3 Log File Locations

Log File	Location
Java Enterprise System installer logs These logs contain information about portal packages installed and basic configuration status.	`/var/sadm/install/logs`
Detailed Portal Server configuration logs	`/var/opt/SUNWportal/logs/config/portal.fabric.0.log`
Portal Admin and Mbean logs	`/var/opt/SUNWportal/logs/admin/portal.0.0.log`
Runtime Portal instance logs	`/var/opt/SUNWportal/portals/`*portalID* `/logs`

Portal Server File Locations

TABLE D–4 Portal Server File Locations

Files	Location
Portal Server packages	`/opt/SUNWportal/`
Host-level configuration files	`/etc/opt`
Instance structure and files	`/var/optSUNWportal` contains the following: ■ `derby` ■ `logs` ■ `portals/`*portal-ID* ■ `Communitytemplates` ■ `config` ■ `desktop` ■ `portletapps` ■ `War` ■ `Web-src` ■ `wiki` ■ `searchservers/`*server-ID*

Community Sample Files

The community Sample Files allow you to set up and configure a community. The following sections list the locations of the community sample files:

Community Template Locations

The community templates are located in `/var/opt/SUNWappserver/domains/domain1/applications/j2ee-modules/portal/communitytemplates`. The following list shows the template directories and property file names for the community sample:

- `2column/` — Directory location for the two-column template contents.
- `2column.properties` — Properties file for the two–column template.
- `menucontainer/` — Directory location for the left-side navigation template contents.
- `menucontainer.properties` — Properties file for the left-side navigation template.
- `wiki/` — Directory location for the wiki template contents.
- `wiki.properties` — Properties file for the wiki template.

Dynamic Content File Locations

Dynamic Content files are located in `/var/opt/SUNWportal/portals/portal1/desktop/community_sample/`.

- `tld/portletSetupTags.tld` — Provides single tag to include Lockhart theme support for stylesheet and javascript functionality.
- `message.properties` — Provides properties used in error/*.template files.
- `openURLInParent.js` — Provides javascript to allow for detached portlets.
- `singlePreferenceMenubar.jsp` — Provides menubar used by JSPDynamicSingleContainer.
- `singlePreferenceHeader.jsp` — Provides header used by JSPDynamicSingleContainer.
- `header.jsp` — Provides header used by all other Containers.
- `footer.html` — Provides footer used by all Containers.
- `error` — Provides error templates.
- `datetime.jsp` — Provides for date/time in header.
- `breadcrumb.jsp` — Provides for breadcrumb in headers.
- `SearchProvider` — Provides for Community look and feel changes.
- `portletEdit.jsp` — Wraps portlet and Provider content in edit page.

- PagePreferencesContainer — Provides for changing Container content and layout preferences
- Login — Provides for login and account creation.
- JSPTableContainerProvider — Provides for portlets in row/column layout.
- JSPMenuContainerProvider — Provides for left-side navigation layout.
- JSPEditContainer — Provides for portlet and Provider preference editing.
- JSPDynamicSingleContainer — Used by Discussions.
- DiscussionProvider — Provides for Community look and feel changes.
- CommunityParentContainer — Instance of JSPDynamicSingleContainer for Community Sample .
- AccountPreferencesContainer — Provides for changing user preferences

Community Sample Portal Properties File Location

The CommunitySamplePortal.properties file contains the properties settings for the community sample. This file is located in /var/opt/SUNWportal/portals/portal1/desktop/classes/CommunitySamplePortal.properties. Note: changing properties requires a server restart.

Note – If you change the community sample properties requires you to restart the server.

Static File Locations

The static content for the community sample is located in /var/opt/SUNWappserver/domains/domain1/applications/j2ee-modules/portal/community/.

- js/ — JavaScript used, including drop-down menu.
- images/desktop/*.gif — non Lockhart images.
- css/desktop.css — non Lockhart stylesheet.
- content/
  - Documentation/index.html — html scrapped for Documentation channel.
  - Introduction/index.html — html scrapped for Introduction channel.
  - Help/*.html — Community Sample help files.
  - Help/graphics/*.gif — Community Sample help file.

Comparison of Solaris and Linux Path Names

TABLE D–5 Comparison of Solaris and Linux Path Names

Solaris Path Name	Linux Path Name
`/opt/SUNWportal` (default)	`/opt/sun/portal` (default)
`/etc/opt/SUNWportal` (config)	`/etc/opt/sun/portal` (config)
`/var/opt/SUNWportal` (data)	`/var/opt/sun/portal` (data)

Index

A

accelerators, Gateway, 59
access control
  Gateway, 59
  limiting, 51
  NetFile, 63-64
  Netlet, 62
  role-based, 51
Access Control Instructions, 106
Access Manager
  administration console, 55-56
  cache and sessions, 104
  components, 55-56
  customizing, 106
  description, 23
  description and benefits, 23
  organization tree, 105
  single sign-on, 55-56
Access Manager SDK, components, 67
administration console tasks, 55-56
aggregation
  description and benefits, 28
  strategy, 107
Allowed URLs and Denied URLs lists, Gateway, 59
`amSDKStats` log, 104
anonymous Desktop, 108
applets, NetFile, 63
application servers
  clustering, 146-149
  requirements, 95
  support for, 145
applications
  degree of integration, 113
  dynamic port, 60-61
  integrating, 112-113
  portal, 112
  static port, 60-61
  third-party, 112
authentication
  Access Manager, 108
  basic, 83
  custom, 112
  Gateway, 58
  LDAP, 23, 69
  modes, 58
  Personal Digital Certificate (PDC), 59
  policy based, 23
  Portal Server, 50
  server, 70
  UNIX, 63
average session time, 38
average time between page requests, 37

B

back-end servers, 41
banner, 56
baseline portal performance analysis, 103
basic authentication, 58
BEA WebLogic, 147-148
bottlenecks, building modules, 79

building modules
  deploying, 79-80
  description, 73
  Directory Server, 77
  high availability and, 73-78
  Search service, 80
  transparent failover, 77-78
business-to-consumer portal, 99
business-to-employee portal, 99

C
cache hit ratio, 104
caching appliance, and reverse proxy, 56
channels
  description, 112
  organizing content, 111
checkpointing mechanisms, 74
chroot environment, 57
Citrix, 33
client detection API, 110
client support, 110-111
clustering
  application servers, 145, 146-149
  session failover, 145
communication links, 68
community deployments
  databases for, 115-130
  Derby and multiple instances, 117-124
  Oracle and high availability, 124-130
components
  Access Manager Server, 55-56
  NetFile, 63
  Portal Server, 55-56
  Secure Remote Access, 57-66
concurrent sessions, 36-37, 37-38
concurrent users, 37-38
configuration data, 109
configuring, HTTP proxy, 156-157
content, placing, 111
CPUs
  estimating number, 36
  Gateway instances, 43
  vertical scaling, 49

credentials, NetFile, 63
customizing
  Access Manager service, 106
  affects on performance, 95
  baseline figures, 95

D
data centers, sizing and, 96
delegated administration, 24
Demilitarized Zone (DMZ), description, 51
Demilitarized Zone (DMZ), description, 55-56
deployment
  bottlenecks, 79
  building modules, 75, 79
  communities, 115-130
  ISP hosting, 71
  providers, 108
  scenarios
    no single point of failure, 75
    Secure Remote Access, 75
    single node, 75
    transparent failover, 77-78
Derby, 117-124
designing
  for localization, 51
  security strategies, 49-51
  use case scenarios, 45-47
Desktop type, 40
directories
  installed for Portal Server, 171-172
  installed for Secure Remote Access, 172
Directory Information Tree, 105, 106
directory replica, 77
Directory Server
  building modules, 77
  clustering, 74
  description, 55-56
  structure design, 105-106
Directory service, description, 23
directory structure, Secure Remote Access, 172
Discussion channel, 26
display profile
  and JSP files, 108

display profile *(Continued)*
content design, 111
DTD location, 172
extracting, 156
properties, 51
reloading, 156
troubleshooting, 155-156
DIT, 105
documenting, portal functions, 102
`dp-org.xml` file, 172
`dpadmin` command, 156
dynamic port applications, 60-61
dynamic web applications, 109

E
encryption
See also security
128-bit, 83
40-bit, 59
Netlet, 60
Portal Server, 50
Enterprise JavaBeans, 95
error logging, levels, 103
`/etc/opt/SUNWps` directory, 171
example use case, Authenticate Portal User, 46-47
extracting the display profile, 156

F
failover
Directory Server, 48
session, 74
file compression, NetFile, 64
FTP, NetFile, 62

G
Gateway
accelerators, 59
access control, 59
Allowed URLs and Denied URLs, 59

Gateway *(Continued)*
and HTTP basic authentication, 58
and Non Authenticated URL, 59
authentication, 58
chroot environment, 57
high availability, 48
HTTP and HTTPS, 57
logging, 59
multiple instances, 57-58
Netlet traffic, 59
overview, 57-59
page configuration, 39-40
profile, 58
proxies, 58, 70
session
information, Gateway, 59
stickiness, 58
SSL, 58-59

H
hardware redundancy, 74, 75
heap size, 98
high availability
building modules and, 73-78
Portal Server components and, 48
high-level architecture, typical installation, 71
high-level portal design, overview, 54
horizontal scaling, description, 49
HTTP and HTTPS modes, and Gateway, 57
HTTP basic authentication, 58
HTTP proxy, configuring, 156-157
HttpSession failover, 74

I
IBM WebSphere Application Server, overview, 149
Identity management, features and benefits, 22-24
independent software vendors, types, 112-113
installing, as a regular user, 50-51
integrating applications, 112-113
Internet Explorer, 110
ISP hosting deployment, 71

`isp` organization, 105
ISVs, types, 112-113

J
Java compatibility, 110
Java properties files, 172
JavaScript
  in Rewriter, 65
  Portal Server Desktop, 56
JavaServer Pages, 108
JAXP, 55-56
JCA, and Refine, 95
jCIFS, NetFile, 62
JDBC, sizing, 95
JSP template files, location, 172
JSPProvider, 108

L
LDAP
  authentication, 69
  transaction numbers, 95
LDIF file, 106
legacy servers, 67
load balancing
  high availability, 74
  Portal Server failures, 76
  Rewriter and, 66
  Secure Remote Access, 48, 77
`locale` file, 172
localization, 51
log files
  location, 171
  Secure Remote Access, 160
  troubleshooting, 154
logging
  errors, 103
  Gateway, 59
  number of active sessions, 104
login type, 39
LoginProvider, 108
low-level portal design, overview, 54

M
Microsoft Exchange
  integrating, 112
  Netlet and, 61
  Netlet Proxy, 62
MIME types,NetFile, 64
monitoring, active sessions, 104
multi-master
  configuration, 74, 79
  Directory Server, 74
multiple network connections, Portal Server, 61
multithreading, NetFile, 65

N
NetFile
  access control, 63-64
  Allowed URLs or Denied URLs, 64
  applet, 63
  components, 63
  compression, 64
  compression types, 64
  initialization, 63
  multithreading, 65
  overview, 62-65
  Portal Server Desktop, 63
  search, 64
  security, 64
  validating credentials, 63
Netlet
  access control, 62
  application integration, 61
  encryption, 44
  encryption (ciphers), 60
  Microsoft Exchange and, 61
  overview, 60-62
  requests and Gateway, 57
  split tunneling, 61-62
  third-party applications, 61
  traffic, 59
  usage characteristics, 44
Netlet Proxy
  Microsoft Exchange, 62
  overview, 62

Netlet Proxy *(Continued)*
- software crash and, 48
- third-party proxy, 62
- transparent failover, 78

Netscape Communicator, 110
NFS, NetFile, 62, 63
Non Authenticated URL list, and Gateway, 59
Novell domain, 63

O

`/opt/SUNWps` directory, 171
`/opt/SUNWps/sdk` directory, 171
Oracle, 124-130
Outlook client, 61

P

packaging, 109
pcAnywhere, 33
peak numbers, 36-37
performance
- Access Manager cache and sessions, 104
- building modules, 79
- garbage collection, 97-98
- memory consumption, 97-98
- thread usage, 104

Personal Digital Certificate (PDC) authentication, 59
personalization, description and benefits, 26-27
placement of portal content, 111
Portal Desktop
- configuration, 40
- design, 107-108

portal key design task list, 164-170
Portal Server
- Access Manager on different nodes, 67-68
- building modules, 73-80
- client support, 110-111
- communication links, 68
- components, 55-56
- configuration files, 171
- design approach, 94
- directory structure, 171-172

Portal Server *(Continued)*
- documenting functions, 102
- growth, 103
- hardware and applications, 40
- high availability and, 48
- high-level design, 54
- instance and servlets, 69
- load balancers, 76
- logical architecture, 55-56
- low-level design, 54
- multiple network connections, 61
- nodes, 57, 67
- overview, 21
- scalability, 48-49
- Secure Remote Access overview, 22
- security, 50
- troubleshooting, 151-157
- tuning goals, 48
- typical installation, 71

Portal Server Desktop
- JavaScript, 56
- NetFile, 63

portlet, description, 112
profile database server, 70
Provider Application Programming Interface, 28
providers, deployment considerations, 108
proxies
- configuration, 58
- failover, 48
- Gateway, 70

Proxylet, overview, 66
`psdp.dtd` file, 172

Q

questions, user behaviors and patterns, 33

R

reloading the display profile, 156
resource bundles, 51
reverse proxy
- description, 92

reverse proxy *(Continued)*
offloading requests, 56
Rewriter
load balancing, 66
overview, 65-66
rulesets, 65
Rewriter Proxy
overview, 66
software crash, 48
robot, 26
roles, 106
rulesets, Rewriter, 65

S
scalability
horizontal, 48-49
portal channels, 107
Secure Remote Access, 43
vertical, 48-49
SDK, description, 110
search, NetFile, 64
Search service
description and benefits, 26
functions, 39
robot, 26
sizing factors, 39
structure, 80
`searchURL` property, 80
Secure Remote Access
components, 57-66
debugging, 157-160
directory structure, 172
features and benefits, 24
load balancing, 48, 77
log files, 160
NetFile, 64
overview, 22
session characteristics, 44
troubleshooting, 157-160
Secure Sockets Layer (SSL), encryption, 49-51
security
NetFile, 64
UNIX, 50
security strategies, 49-51
servlets, and communication, 69
session
characteristics, Secure Remote Access, 44
monitoring, 104
stickiness, 58
session failover
BEA, 148
clustering and, 145
high availability, 74
session information, 59
SimpleWebServiceProvider, 108
single sign-on
Access Manager, and, 50
description, 23
sizing
customizing, 95
JCA and, 95
JDBC, 95
refining, 96-97
search engine factors, 39
tool, 43-44
validating, 96
software
categories, 109-110
packaging, 109
software crash, 48
Solaris Operating Environment, minimizing size of installation, 49-51
split tunneling, 61-62
SRA, and reverse proxy, 92
SSL
Gateway, 58-59
modes, 58
v2 and v3, 58
state data, and Portal Server services, 70
static port applications, 60-61
static portal content, 111
static web content, 109
Subscription channel, 26
Sudo, 51
Sun Cluster software, 74
Sun Enterprise Midframe Line, support for multiple hardware domains, 98-99

Sun Java System Application Server, overview, 147
`SuperAdmin Role`, 106
system capacity, 42
system performance, 42

T
tag library definitions, 172
task list, 164-170
third-party applications
  description, 112
  Netlet, 61
third-party proxy, Netlet Proxy, 62
thread usage, 104
transaction time, 41
transparent failover, building module and, 77-78
troubleshooting
  display profile, 155-156
  Portal Server, 151-157
  Secure Remote Access, 157-160
  Unix processes, 153-154
tuning, goals, 48
tunneling, 61-62

U
Unix, authentication, 63
UNIX, user installation, 50-51
UNIX processes, troubleshooting, 153-154
URL
  access lists, 64
  NetFile, 64
use case scenarios
  designing, 45-47
  example, 46-47
user behaviors and patterns, 33

V
vertical scaling, description, 49
VPN, 24
VPN client, 61

W
WAR file
  and application servers, 146
  contents, 109
  to deploy software, 109
workload conditions, 41-42
worksheets, 161-170

X
XMLProvider, 108