



Sun Java™ System

Portal Server 6 Deployment Planning Guide

2004Q2

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

Part No: 817-5321

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 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 of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

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 in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo 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 architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) 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 service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical 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 © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L'AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit pourront être dérivées des systèmes 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, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE 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.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) 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.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Contents

List of Figures	9
List of Tables	11
About This Guide	13
Who Should Read This Book	13
What You Need to Know	13
How This Book Is Organized	14
Document Conventions Used in This Guide	15
Monospaced Font	15
Italicized Font	15
Square or Straight Brackets	16
Command-Line Prompts	16
Where to Find Related Information	16
Where to Find This Guide Online	16
Sun Welcomes Your Comments	17
Chapter 1 Portal Server Architecture	19
What is a Portal?	19
Types of Portals	20
Collaborative Portals	20
Business Intelligence Portals	21
Portal Server Capabilities	21
Sun Java System Portal Server	22
Sun Java System Secure Remote Access	23
Portal Server in Open Mode	23
Portal Server in Secure Mode	24
Security, Encryption, and Authentication	26
Portal Server Deployment Components	26
Portal Server Architecture	27

Identity Management	28
Portal Server Software Deployment	29
Software Packaging	29
Software Categories	29
Java Compatibility	30
A Typical Portal Server Installation	31
Chapter 2 Portal Server Secure Remote Access Architecture	35
SRA Gateway	36
Multiple Gateway Instances	36
Proxy Configuration	37
Gateway and HTTP Basic Authentication	37
Gateway and SSL Support	38
Gateway Access Control	39
Gateway Logging	39
Using Accelerators with the Gateway	39
Netlet	39
Static and Dynamic Port Applications	40
Netlet and Application Integration	41
Split Tunneling	41
Netlet Proxy	42
NetFile	42
Components	42
Initialization	43
Validating Credentials	43
Access Control	44
Security	44
Special Operations	44
NetFile and Multithreading	45
Rewriter	46
Rewriter Proxy	46
Proxylet	47
Chapter 3 Identifying and Evaluating Your Business and Technical Requirements	49
Business Objectives	49
Technical Goals	51
Mapping Portal Server Features to Your Business Needs	52
Identity Management	52
SRA	54
Search Engine	55
Personalization	56
Aggregation and Integration	57

Understanding User Behaviors and Patterns	58
Chapter 4 Pre-Deployment Considerations	59
Determine Your Tuning Goals	59
Portal Sizing Tips	60
Establish Performance Methodology	60
Portal Sizing	61
Establish Baseline Sizing Figures	62
Customize the Baseline Sizing Figures	67
Validate Baseline Sizing Figures	68
Refine Baseline Sizing Figures	69
Validate Your Final Figures	70
SRA Sizing	70
Identifying Gateway Key Performance Requirements	71
Advanced Gateway Settings	73
SRA Gateway and SSL Hardware Accelerators	74
SRA and Sun Enterprise Midframe Line	75
Chapter 5 Creating Your Portal Design	77
Portal Design Approach	77
Overview of High-Level Portal Design	78
Overview of Low-Level Portal Design	79
Logical Portal Architecture	79
Portal Server and Scalability	81
Vertical Scaling	81
Horizontal Scaling	81
Portal Server and High Availability	82
System Availability	83
Degrees of High Availability	83
Achieving High Availability for Portal Server	83
Portal Server System Communication Links	84
Working with Portal Server Building Modules	87
Building Modules and High Availability Scenarios	88
Building Module Constraints	95
Deploying Your Building Module Solution	95
Designing Portal Use Case Scenarios	97
Elements of Portal Use Cases	98
Example Use Case: Authenticate Portal User	99
Designing Portal Security Strategies	100
Securing the Operating Environment	100
Using Platform Security	101
Using a Demilitarized Zone (DMZ)	102

Portal Server and Identity Server on Different Nodes	103
Designing SRA Deployment Scenarios	108
Basic SRA Configuration	109
Disable Netlet	110
Proxylet	111
Multiple Gateway Instances	112
Netlet and Rewriter Proxies	113
Netlet and Rewriter Proxies on Separate Nodes	115
Using Two Gateways and Netlet Proxy	116
Using an Accelerator	117
Netlet with 3rd Party Proxy	118
Reverse Proxy	119
Designing for Localization	120
Content and Design Implementation	120
Integration Design	121
Identity and Directory Structure Design	124
Implementing Single Sign-On	125
Portal Desktop Design	125
Client Support	128
Chapter 6 The Production Environment	129
Moving to a Production Environment	129
Monitoring and Tuning	129
Documenting the Portal	130
Monitoring Portal Server	131
Memory Consumption and Garbage Collection	132
CPU Utilization	133
Identity Server Cache and Sessions	133
Thread Usage	134
Portal Usage Information	134
Appendix A Installed Product Layout	135
Directories Installed for Portal Server	135
Directories Installed for SRA	136
Configuration Files	137
Appendix B Analysis Tools	139
mpstat	140
iostat	142
netstat	143
Tuning Parameters for <code>/etc/system</code>	146

Appendix C Portal Server and Application Servers	149
Introduction to Application Server Support in Portal Server	149
Portal Server on an Application Server Cluster	150
Overview of Sun Java System Application Server	151
Overview of BEA WebLogic Server Clusters	151
Overview of IBM WebSphere Application Server	153
Appendix D Troubleshooting Your Portal Deployment	155
Troubleshooting Portal Server	155
UNIX Processes	155
Log Files	156
Recovering the Search Database	156
Working with the Display Profile	156
High CPU Utilization for Portal Server Instance	157
Configuring a Sun Java System Portal Server Instance to Use an HTTP Proxy	158
Troubleshooting SRA	158
Debugging the Gateway	158
Introduction to shooter	159
Using shooter	160
SRA Log Files	161
Appendix E Portal Deployment Worksheets	163
Portal Assessment Worksheets	163
Portal Key Design Task List	167

List of Figures

Figure 1-1	Portal Server in Open Mode	24
Figure 1-2	Portal Server in Secure Mode	25
Figure 1-3	High-level Architecture for a Business-to-Employee Portal	32
Figure 1-4	SRA Deployment	33
Figure 5-1	Portal Server Communication Links	85
Figure 5-2	Portal Server Building Module Architecture	87
Figure 5-3	Best Effort Scenario	90
Figure 5-4	No Single Point of Failure Example	91
Figure 5-5	Transparent Failover Example Scenario	94
Figure 5-6	Portal Server and Identity Server on Different Nodes	104
Figure 5-7	Two Portal Servers and One Identity Server	105
Figure 5-8	One Portal Server and Two Identity Servers	106
Figure 5-9	Two Portal Servers and Two Identity Servers	107
Figure 5-10	Basic SRA Configuration	109
Figure 5-11	Disable Netlet	110
Figure 5-12	Proxylet	111
Figure 5-13	Multiple Gateway Instances	112
Figure 5-14	Netlet and Rewriter Proxies	114
Figure 5-15	Proxies on Separate Nodes	115
Figure 5-16	Two Gateways and Netlet Proxy	116
Figure 5-17	SRA Gateway with External Accelerator	117
Figure 5-18	Netlet and Third-Party Proxy	118
Figure 5-19	Using a Reverse Proxy in Front of the Gateway	119

List of Tables

Table 3-1	Identity Management Features and Benefits	52
Table 3-2	SRA Features and Benefits	54
Table 3-3	Search Features and Benefits	55
Table 3-4	Personalization Features and Benefits	56
Table 3-5	Aggregation Features and Benefits	57
Table 5-1	Portal Server High Availability Scenarios	89
Table 5-2	Use Case: Authenticate Portal User	99
Table A-1	Portal Server Directories	135
Table A-2	Portal Server, SRA Directories	136
Table B-1	Performance Analysis Tools	139
Table B-2	/etc/system Options	146
Table B-3	TCP/IP Options	146
Table E-1	General Questions	163
Table E-2	Organizational Questions	164
Table E-3	Business Service-level Expectations Questions	165
Table E-4	Content Management Questions	165
Table E-5	User Management and Security Questions	166
Table E-6	Business Intelligence Questions	166
Table E-7	Architecture Questions	166
Table E-8	Key Design Task List	167

About This Guide

This guide explains how to plan for and deploy Sun Java™ System Portal Server 6 2004Q2 software (formerly Sun ONE Portal Server). Portal Server provides a platform to create portals for your organization's integrated data, knowledge management, and applications. The Portal Server platform offers a complete infrastructure solution for building and deploying all types of portals, including business-to-business, business-to-employee, and business-to-consumer.

This preface includes the following sections:

- [Who Should Read This Book](#)
- [What You Need to Know](#)
- [How This Book Is Organized](#)
- [Where to Find Related Information](#)
- [Where to Find This Guide Online](#)

Who Should Read This Book

You should read this guide if you are responsible for deploying Portal Server at your site.

What You Need to Know

Before you deploy Portal Server, you must be familiar with the following concepts:

- Sun Java™ Enterprise System
- Basic Solaris™ administrative procedures

- Sun Java System Identity Server
- Sun Java System Directory Server
- Sun Java System Web Server
- JavaServer Pages™ technology
- LDAP
- HTML
- XML

How This Book Is Organized

This book contains the following chapters and appendices:

- About This Guide (this chapter)
- [Chapter 1, “Portal Server Architecture”](#)

This chapter describes the basic architecture of the Portal Server product.

- [Chapter 2, “Portal Server Secure Remote Access Architecture”](#)

This chapter describes the Portal Server Secure Remote Access architecture, including the key components of Secure Remote Access with respect to their role in providing secure remote access to corporate intranet resources from outside the intranet.

- [Chapter 3, “Identifying and Evaluating Your Business and Technical Requirements”](#)

This chapter describes how to analyze your organization’s needs and requirements that lead to designing your portal deployment.

- [Chapter 4, “Pre-Deployment Considerations”](#)

This chapter describes how to establish a baseline sizing figure for your portal. With a baseline figure established, you can then refine that figure to account for scalability, high availability, reliability, and good performance.

- [Chapter 5, “Creating Your Portal Design”](#)

This chapter describes how to create your high-level and low-level portal design and provides information on creating specific sections of your design plan.

- [Chapter 6, “The Production Environment”](#)
This chapter describes how to tune and monitor your portal.
- [Appendix A, “Installed Product Layout”](#)
This appendix describes the directories and configuration files for Portal Server and Sun Java™ System Portal Server Secure Remote Access (SRA).
- [Appendix B, “Analysis Tools”](#)
This appendix describes analysis tools for tuning the operating system.
- [Appendix C, “Portal Server and Application Servers”](#)
This appendix describes the support for application servers.
- [Appendix D, “Troubleshooting Your Portal Deployment”](#)
This appendix describes how to troubleshoot the Sun Java™ System Portal Server software and the Sun Java™ System Portal Server Secure Remote Access (SRA) product.
- [Appendix E, “Portal Deployment Worksheets”](#)
This appendix provides various worksheets to help in the deployment process.

Document Conventions Used in This Guide

Monospaced Font

`Monospaced font` is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

Italicized Font

An *italicized font* is used to represent text that you enter using information that is unique to your installation (for example, variables). It is used for server paths and names and account IDs.

Square or Straight Brackets

Square (or straight) brackets `[]` are used to enclose optional parameters. For example, in the Portal Server documentation, you will see the usage for the `dpadmin` command described as follows:

```
dpadmin list|modify|add|remove [command-specific options]
```

The presence of `[command-specific]` indicates that there are optional parameters that can be added to the `dpadmin` command.

Command-Line Prompts

Command-line prompts (for example, `%` for a C-Shell, or `$` for a Korn or Bourne shell) are not displayed in the examples. Depending on which operating system environment you are using, you will see a variety of different command-line prompts. However, you should enter the command as it appears in the document unless specifically noted otherwise.

Where to Find Related Information

Use the following URL to view all the Portal Server documentation:

<http://docs.sun.com>

Listed below are the additional documents that are available:

- *Portal Server 6 Administration Guide*
- *Portal Server 6 Developer's Guide*
- *Portal Server 6.2 Migration Guide*
- *Portal Server Secure Remote Access 6 Administration Guide*

Where to Find This Guide Online

You can find this guide online in PDF and HTML formats at the following URL:

<http://docs.sun.com>

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. Use the web-based form to provide feedback to Sun:

<http://www.sun.com/hwdocs/feedback>

Please provide the full document title and part number in the appropriate fields. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the part number for this document is 817-5321-10.

Sun Welcomes Your Comments

Portal Server Architecture

This chapter contains the following sections:

- [What is a Portal?](#)
- [Types of Portals](#)
- [Portal Server Capabilities](#)
- [Sun Java System Portal Server](#)
- [Sun Java System Secure Remote Access](#)
- [Security, Encryption, and Authentication](#)
- [Portal Server Deployment Components](#)
- [Portal Server Architecture](#)
- [Identity Management](#)
- [A Typical Portal Server Installation](#)

What is a Portal?

Portals provide the user with a single point of access to a wide variety of content, data, and services throughout an enterprise. The content displayed through portal providers, channels, and portlets on the portal page can be personalized based on user preferences, user role or department within an organization, site design, and marketing campaigns for customers as end-users.

Portals serve as a simple, unified access point to web applications. Portals also do much more—they provide valuable functions like security, search, collaboration, and workflow. A portal delivers integrated content and applications, plus a unified, collaborative workplace. Indeed, portals are the next-generation desktop,

delivering e-business applications over the web to all kinds of client devices. A complete portal solution should provide users with convenient access to everything they need to get their tasks done—any time, anywhere, in a secure manner.

Types of Portals

With many new portal products being announced, the marketplace has become very confusing. Indeed, any product or application that provides a web interface to business content could be classified as a portal. For this reason portals come in many flavors and have many different uses. They could be classified as one of the following:

- [Collaborative Portals](#)
- [Business Intelligence Portals](#)

Collaborative Portals

Collaborative portals help business users organize, find, and share unstructured office and groupware content—for example, e-mail, discussion group material, office documents, forms, memos, meeting minutes, web documents, and some support for live feeds. They differ from Internet and intranet portals not only in supporting a wider range of information, but also by providing a rich set of content management and collaborative services.

Content management services include text mining and clustering of related unstructured information, information categorization to classify it and make it easy to find, summarization to generate abstracts for documents, publish and subscribe, finding people, and tracking expertise.

Collaborative services allow users to chat, organize meetings, share calendaring information, define user communities, participate in net meetings, and share information in discussion groups and on white boards. Collaborative portals are mainly used internally as a corporate facility.

Business Intelligence Portals

Business intelligence portals provide executives, managers, and business analysts with easy access to business intelligence for making key business decisions. This type of portal typically indexes business intelligence reports, analyses, and canned queries, and are associated with financial management, customer relationship management, and supply chain performance management. They also provide seamless access to business intelligence tools (reporting, OLAP, data mining), and packaged analytic applications. They also support alerting, publishing and subscribing. Peoplesoft is a typical vendor provider of business intelligence types of portal.

Types of business intelligence portals include:

- Procurement portal
- Self-service portal
- Business portal
- e-Commerce portal
- Sales support
- Customer relationship management, operations, and employee portals
- Consumer portal

Portal Server Capabilities

Sun Java™ System Portal Server 6 2004Q2 software (formerly Sun ONE Portal Server) provides the following capabilities to your organization:

- Secure access and authorized connectivity, optionally using encryption between the user's browser and the enterprise
- Authentication of users before allowing access to a set of resources that are specific for each user
- Support for abstractions that provide the ability to pull content from a variety of sources and aggregate and personalize it into an output format suitable for the user's device
- A search engine infrastructure to enable intranet content to be organized and accessed from the portal
- Ability to store user- and service-specific persistent data

- Access to commonly needed applications for accessing services such as mail, calendar, and file storage
- An administration interface enabling delegated and remote administration
- Single sign-on and security features, enabling standard access to enterprise applications and content
- Personalization through the use of portal providers, portlet and web service remote portlet (WSRP).
- Publishing and managing content (provided by third-party applications such as FatWire, pCM)

Sun Java System Portal Server

Portal Server is a component of the Sun Java™ Enterprise System. Sun Java Enterprise System is a software system that 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 key 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 robust business-to-employee, business-to-business, and business-to-consumer portals. The Sun Java™ System Portal Server Secure Remote Access (SRA) (formerly Sun ONE Portal Server, Secure Remote Access) product provides additional 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. The optimal deployment for each enterprise depends on the types of applications that Java Enterprise System is supporting, 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 as long as the software is an appropriate version.

Sun Java System Secure Remote Access

Sun Java System Portal Server Secure Remote Access (SRA) offers browser-based secure access to portal content and services from any remote browser enabled with Java technology.

SRA is a cost-effective, secure access solution that is accessible to users from any Java™ technology-enabled browser, eliminating the need for client software. Integration with Portal Server software ensures that users receive secure encrypted access to the content and services that they have permission to access.

SRA is targeted toward enterprises deploying highly secure remote access portals. These portals emphasize security, protection, and privacy of intranet resources. The SRA services—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 runs in open mode and secure mode, that is, either without SRA or with SRA.

Portal Sever in Open Mode

In open mode, Portal Server is installed without SRA. The typical public portal, such as `my.yahoo`, runs without secure access using only the HTTP protocol. Although you can configure Portal Server to use the HTTPS protocol in open mode (either during or after installation), secure remote access is not possible. This means that users cannot access remote file systems and applications.

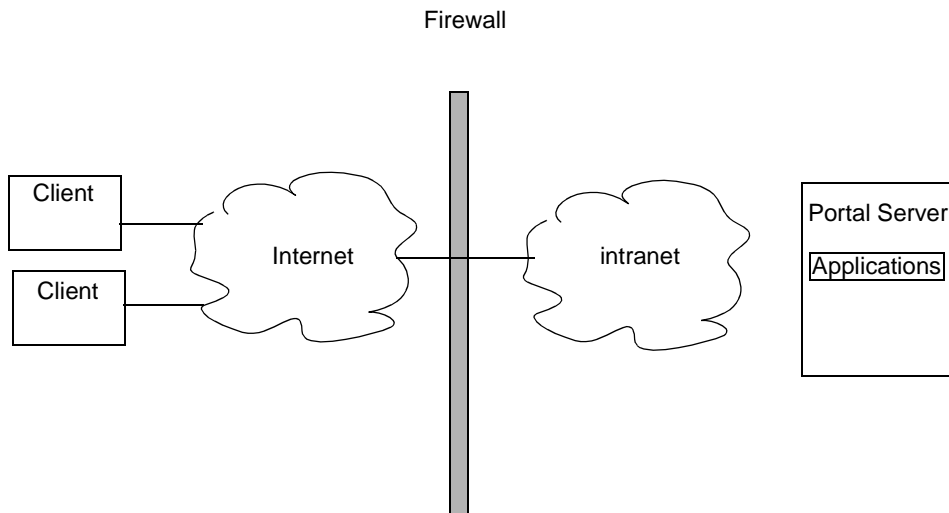
The main difference between an open portal and a secure portal is that the services presented by the open portal typically reside within the demilitarized zone (DMZ) and not within the secured intranet.

If the portal does not contain sensitive information (deploying public information and allowing access to free applications), then responses to access requests by a large number of users is faster than secure mode.

[Figure 1-1](#) shows Portal Server configured for open mode. In this figure, Portal Server is installed on a single server behind the firewall. Multiple clients access the Portal Server system across the Internet through the single firewall, or from a web proxy server that sits behind a firewall.

NOTE You can provide secure access to users of web-enabled resources by running Portal Server in open mode with the HTTPS protocol. However, without SRA, you cannot provide secure remote access to file systems or TCP/IP applications.

Figure 1-1 Portal Server in Open Mode



Portal Server in Secure Mode

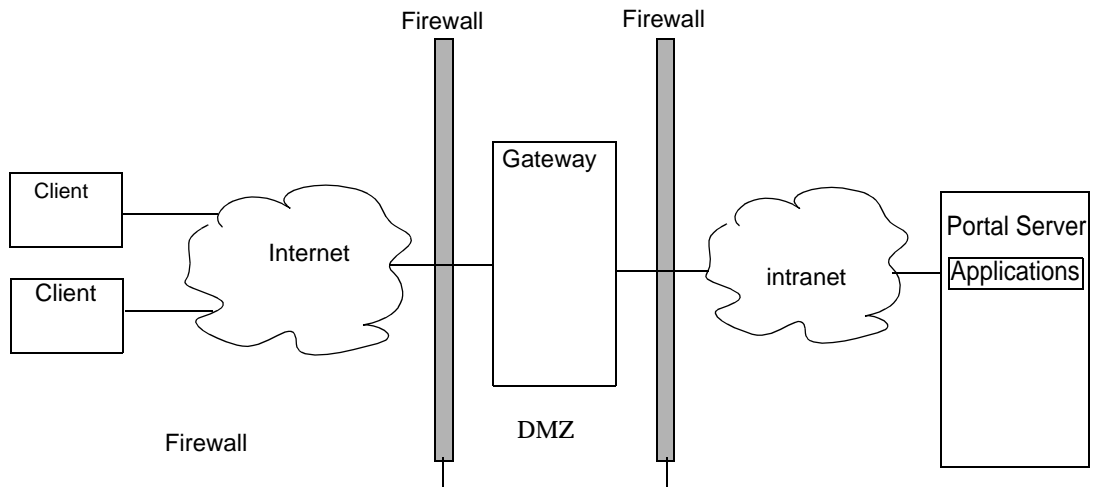
In secure mode, Portal Server is installed with SRA. Secure mode provides users with secure remote access to required intranet file systems and applications.

The main advantage of SRA is that only the IP address of the Gateway is published to the Internet. All other services and their IP addresses are hidden and never published to a Domain Name Service (DNS) that is running on the public network (such as the Internet).

The Gateway resides in the demilitarized zone (DMZ). The Gateway provides a single secure access point to all intranet URLs and applications, thus reducing the number of ports to be opened in the firewall. All other Sun Java System services such as Session, Authentication, and Portal Desktop, reside behind the DMZ in the secured intranet. Communication from the client browser to the Gateway is encrypted using HTTP over Secure Sockets Layer (HTTPS). Communication from the Gateway to the server and intranet resources can be either HTTP or HTTPS.

Figure 1-2 shows Portal Server installed with SRA. SSL is used to encrypt the connection between the client and the Gateway over the Internet. SSL can also be used to encrypt the connection between the Gateway and the Portal Server system. The presence of a Gateway between the intranet and the Internet extends the secure path between the client and the Portal Server system.

Figure 1-2 Portal Server in Secure Mode



You can add additional servers and Gateways for site expansion. You can also configure the components of SRA in various ways based on your business requirements.

Security, Encryption, and Authentication

Portal Server system security 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 Identity Server policy to enforce URL-based access policy.

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

Another layer of security is provided by SRA. It uses HTTPS by default for connecting the client browser to the intranet. The Gateway uses Rewriter 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.

Portal Server Deployment Components

Portal Server deployment consists of the following components:

- Sun Java™ System Identity Server software

Sun Java System Identity Server 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 provides the Java run-time environment for all Java software in Portal Server and its underlying components. Portal Server depends on the JDK of the web container.
- Network Security Services for Java™

- Sun Java™ System Web Server (JSS),
- Java™ API for XML Processing (JAXP),
- Sun Java™ System Directory Server software

Sun Java System 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

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

- BEA WebLogic Server™
- IBM WebSphere® Application Server

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

NOTE See the *Portal Server 6 Release Notes* for specific versions of products supported by Portal Server.

Portal Server Architecture

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 server where Portal Server resides, Sun Java System Web Server or other web container. You can also install the Search component on this node if desired. Identity Server can reside here.
- **Identity Server node**—The server where Identity Server can reside. Identity Server 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 upon which you install the SRA Gateway. You can also install the Gateway on the portal node, though in general, because you locate the Gateway in the DMZ, it will be on a separate, non-portal node.
- **Netlet Proxy node**—Optional. The node used to run applications securely between users' remote desktops and the servers running applications on your intranet.
- **Rewriter Proxy node**—Optional. The node used to run applications securely between users' remote desktops and the servers running applications on your intranet.
- **Directory Server**—The server running Sun Java System Directory Server software. You can install Directory Server on a separate node.
- **Other servers**—These servers, such as mail, file, and legacy servers, provide backend support, data, and applications to portal users.

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 they serve the organization or company? What do they need to do, and what should they be able to access? How can others help with the administrative work?

Identity Server software consists of the following components:

- Java APIs used to access SSO Token, user profiles, logging, and debugging
- Command line tools such as amadmin, amserver, and ampassword
- Web application services such as session, authentication, logging, and naming
- Administration console web application
- Identity Server SDK
- Identity Server console SDK
- Authentication daemons that support the web applications

See the *Identity Server Deployment Planning Guide* for more information.

Portal Server Software Deployment

This section provides information on software deployed in Portal Server. It provides information on the software packaging mechanism, the software categories within the system, and the Java compatibility of the 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 Sun Java System 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**—Includes Java servlets, JSP files, content providers, and other items that the Java web container processes when accessed by the user’s browser. For Portal Server, these files are installed in the web server.
- **Static web content**—Includes static HTML files, images, applet JAR files, and other items that can be served up directly by the web server without using the Java web container. For Portal Server, these files are also installed in the web server.

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

- **Configuration data**—Includes data that is installed into the directory, that is, the Identity Server 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.
- **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. If a component does not export any public Java APIs, it would not have this package.

Java Compatibility

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 servlets interface except where uses of special interfaces are identified. This includes compatibility with Java 2 and later.
- **Stand-alone Java processes**—Stand-alone Java processes are compatible with Java 2 and later. Some Portal Server software, specifically in SRA, use JNI to call C APIs. These calls are necessary to enable the system to run as the user nobody.

A Typical Portal Server Installation

[Figure 1-3 on page 32](#) 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. See [Chapter 5, “Creating Your Portal Design”](#), for more detailed information on portal design.

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 DMZ along with other systems accessible from the Internet, including web servers, proxy/cache servers, and mail Gateways. The portal node, portal search node, and directory server, are hosted on the internal network where they have access to systems and services ranging from individual employee desktop systems to legacy systems.

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

In [Figure 1-3 on page 32](#), users on the Internet access the Gateway from a browser. The Gateway connects the user to the IP address and port for the portal they are 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 on to the portal node, or directly to the service access on the enterprise internal network.

Figure 1-3 High-level Architecture for a Business-to-Employee Portal

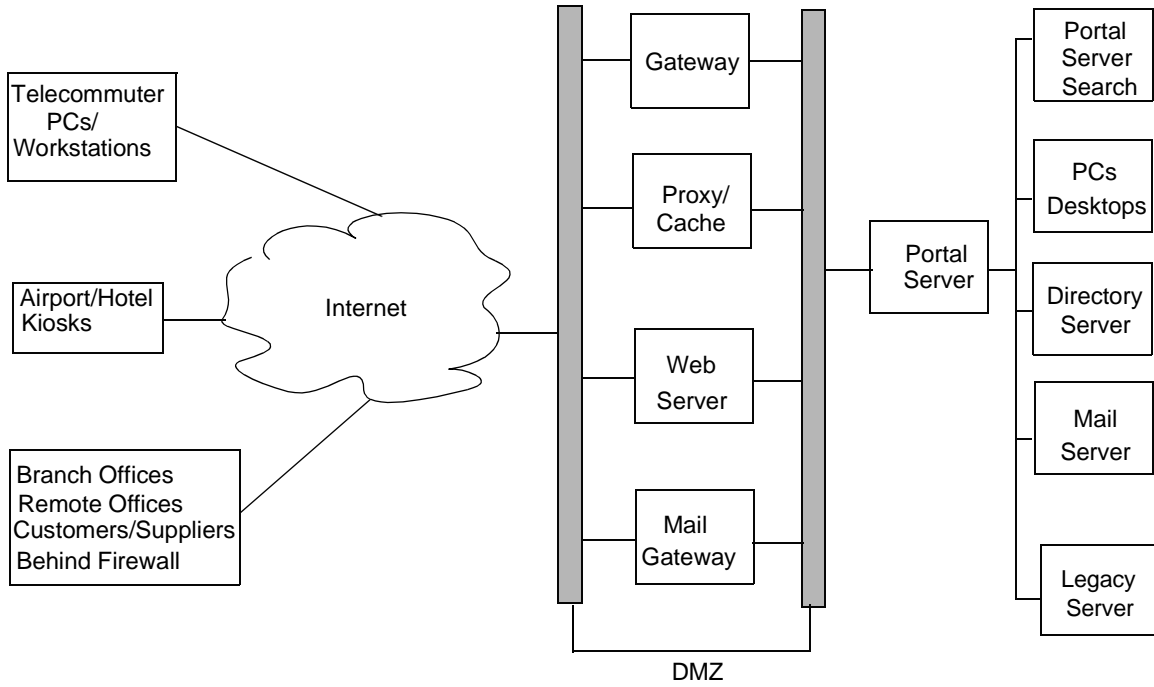
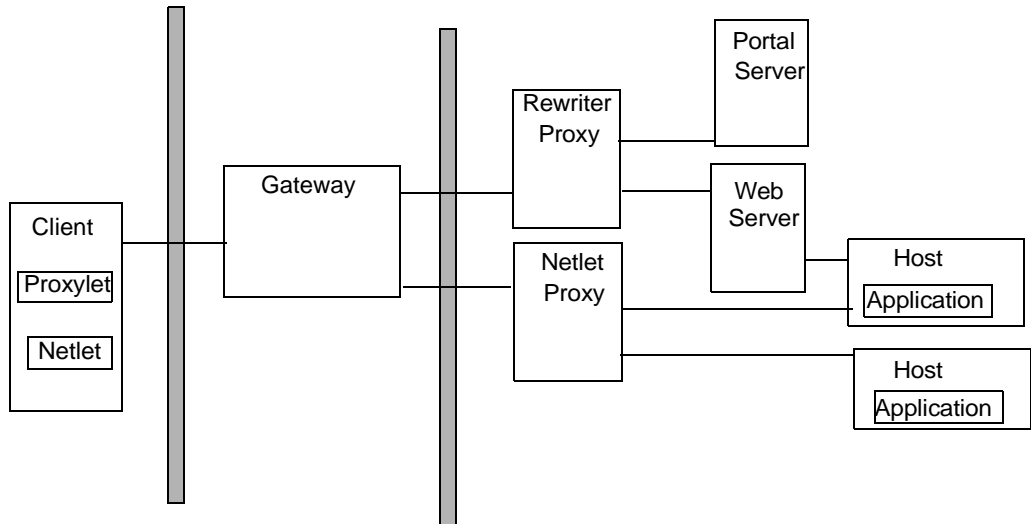


Figure 1-4 shows a Portal Server deployment with SRA services. See [Chapter 2, "Portal Server Secure Remote Access Architecture"](#) for details.

Figure 1-4 SRA Deployment



A Typical Portal Server Installation

Portal Server Secure Remote Access Architecture

This chapter describes the Sun Java™ System Portal Server Secure Remote Access (SRA) (formerly Sun ONE Portal Server, Secure Remote Access) architecture.

Sun Java System Secure Remote Access (SRA) uses Identity Server software to store all its configuration information. Information about the machines (such as host name, IP address), is stored on the host where the Gateway is installed.

You administer the configuration information stored in Sun Java System Identity Server through the Identity Server administration console. This chapter describes the following SRA components:

- [SRA Gateway](#)
- [Netlet](#)
- [Netlet Proxy](#)
- [NetFile](#)
- [Rewriter](#)
- [Rewriter Proxy](#)
- [Proxylet](#)

SRA Gateway

The SRA Gateway is a standalone Java process that can be considered to be stateless, since state information can be rebuilt transparently to the end user. The Gateway listens on the configured ports to accept HTTP and HTTPS requests. Upon receiving a request, the Gateway checks session validity and the 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 will work on the extranet.

All the Gateway configuration information is stored in the Identity Server'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. All machine-specific information is stored in a configuration file in the local file system where the Gateway is installed. This 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 modes, simultaneously. This 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).

You can also run the Gateway in `chroot` environments. See the *Portal Server Secure Remote Access 6 Administration Guide* for more information.

Multiple Gateway Instances

If desired, you can run multiple Gateway instances on a single machine—this is referred as a *multihomed* Gateway. Each Gateway instance listens on separate port(s). 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. In essence, this provides the flexibility of having a different Gateway server certificate for each domain.

When you configure the Gateway with multiple instances of Portal Server, the Gateway automatically performs round-robin load balancing by logging in users with the different servers, alternately. The Gateway also keeps a list of active servers to avoid trying to login users to an inactive server. This mechanism helps to avoid single points of failure with Portal Server.

NOTE Session stickiness is not required in front of a Gateway (unless you are using Netlet), although it is recommended for performance reasons. On the other hand, session stickiness to the Portal Server instances is enforced by SRA.

Proxy Configuration

The Gateway uses proxies that are specified in its profile to retrieve contents from various web servers within the intranet and extranet. It is possible to 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, it is stored as part of the gateway profile that the Gateway uses automatically while 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 Identity Server 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 Identity 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 Identity Server authentication modules. With SSL client authentication, the SSL handshake ends at the Gateway. This PDC-based authentication is integrated along with the Identity Server's certificate-based authentication. Thus, the client certificate is handled by Identity Server 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 Identity Server. 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 validity of the session against the Identity Server 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 Identity 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 can provide 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 Identity Server administration console.

Static and Dynamic Port Applications

Static port applications run on known or static ports where they can be contacted by clients. 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:

- 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.
-

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, they 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. See the *Portal Server Secure Remote Access 6 Administration Guide* for more information.

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)**—There are two NetFile servlets 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 encodings).

NetFile uses Identity Server to store its own profile, as well as user settings and preferences. You administer NetFile through the Identity 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). Because it is possible to have an independent password for each share, users need to enter their credentials for every share (except for common hosts) that you add.

NetFile uses UNIX Authentication from the Identity Server 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 those 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 SRA 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 Identity 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 Identity Server 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 will perform 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. See the *Portal Server Secure Remote Access 6 Administration Guide* for more information.

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

1. From the request, SRA identifies the URI of the intranet page or Internet page that needs to be served.
2. SRA uses the proxy settings to connect to the identified URL.
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, SRA 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 there is a firewall 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.
- 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 (no 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 they 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 JSSE 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 Identity 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. Integration with third-party software such as Microsoft Exchange Server is easy. Also there is an increase in the performance of the Gateway as Proxylet does not deal with web content.

Proxylet

Identifying and Evaluating Your Business and Technical Requirements

The first step in planning your deployment is identifying your Sun Java™ System Portal Server business and technical requirements.. You need to gather both business and technical requirements before you can address architecture and design issues.

This chapter contains the following sections:

- [Business Objectives](#)
- [Technical Goals](#)
- [Mapping Portal Server Features to Your Business Needs](#)
- [Understanding User Behaviors and Patterns](#)

Business Objectives

Your business requirements address your organization's problems and opportunities, and include such factors as:

- Services
- Service availability
- Future growth
- New technologies
- Capital investment

To be useful in formulating design requirements, the business requirements must address detailed goals and objectives.

The business goals of your portal affect deployment decision. It is important to understand your objectives. If you do not understand your business requirements, you can easily make erroneous assumptions that could affect the accuracy of your deployment estimates.

Use these questions to help you identify your business objectives:

- What are the business goals 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 key priorities for the portal? (If you plan to deploy your portal in phases, identify key priorities for each phase.)

(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 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)?

Technical Goals

Your technical requirement (often called functional requirement) discuss the details of your organization's system needs and desired results, and include such factors as:

- Performance
- Security
- Reliability
- Expected performance criteria of the portal

The technical requirements define all functions required of an architecture and provide guidelines for how each component works and integrates to form an entire system. Your organization needs technical requirements to formulate the best design approaches and apply the appropriate technologies to accomplish the desired architectural solution for your portal.

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 level of concurrency—the number of users who can be connected at any given time—is necessary?
- Should access to the portal be through intranet or Internet?
- Will your portal be deployed in one phase, or many phases? (Describe each phase and what will change from phase to phase.)

Mapping Portal Server Features to Your Business Needs

The previous sections posed questions to you about the various areas of the Sun Java System Portal Server system from a high-level perspective of business and technical needs. 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 will assist you in developing a deployment plan in a timely and cost effective manner.

NOTE In all likelihood, your Sun Java System sales representative has previously discussed these topics with you. Thus, 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 they serve the organization or company? What do they need to do, and what should they be able to access? How can others help with the administrative work?

[Table 3-1](#) shows the identity management features and their benefits.

Table 3-1 Identity Management Features and Benefits

Feature	Description	Benefit
Directory service	Sun Java System Portal Server uses Sun Java™ System Identity Server (formerly Sun ONE Identity Server) and Sun Java™ System Directory Server	<p>Portal Server uses an LDAP directory for storing user profiles, roles, and identity information for the purpose of authentication, single sign-on (SSO), delegated administration, and personalization</p> <p>Portal Server uses an open schema that can reside in a centralized user directory, thereby leveraging an enterprise or service provider's investment in the Identity Server and Sun Java System Directory Server products.</p>

Table 3-1 Identity Management Features and Benefits *(Continued)*

Feature	Description	Benefit
User, policy, and provisioning management	Sun Java System Identity Server 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.	<p>Provides a centralized identity management solution for storing and managing identity information, which is integrated with a policy solution to enforce access rights, greatly simplifying these challenges. Extends a common identity to handle new applications, enables applications to share administrative work, and simplifies tasks normally associated with building these services from scratch.</p> <p>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.</p> <p>Provides secure policy-based access to applications. Ensures secure access as portal deployments expand beyond employee LAN access.</p>
Single sign-on (SSO)	Identity Server 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 key content, applications, and services. By being more secure, the more cost-effective and productive your organization and business will be.
Delegated administration	The Identity 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 is a key functionality in portals. 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 those users that are allowed access.

SRA

Table 3-2 shows the Sun Java™ System Secure Remote Access (SRA) (formerly Sun ONE Portal Server) features and their benefits

Table 3-2 SRA 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.	<p>Extends an enterprise’s content, applications, files, and services located behind firewalls to authorized suppliers, business partners, and employees.</p> <p>To prevent denial of service attacks, you can use both internal and external DMZ-based Gateways.</p>
SRA core	<p>Users achieve remote access through four components:</p> <ul style="list-style-type: none"> • Gateway • NetFile • Netlet • Proxylet 	<p>This component has four parts:</p> <ul style="list-style-type: none"> • 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—Redirects a URL to 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.

Table 3-2 SRA Features and Benefits *(Continued)*

Feature	Description	Benefit
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.	Using the Rewriter Proxy enables secure HTTP traffic between the Gateway and intranet computers and offers two advantages: <ul style="list-style-type: none"> • If there is a firewall 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 now secure between the Gateway and the intranet even if the destination server only supports HTTP protocol (no HTTPS).

Search Engine

The Search Engine 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 3-3](#) shows the Search features and their benefits.

Table 3-3 Search Features and Benefits

Feature	Description	Benefit
Search Engine	Enables the retrieval of documents based on criteria specified by the end user.	Saves users time by providing easy 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 easy browsing and retrieval.
Robot	The Search Engine 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).

Table 3-3 Search Features and Benefits *(Continued)*

Feature	Description	Benefit
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.

Personalization

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

Table 3-4 shows the personalization features and their benefits.

Table 3-4 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 those building the web sites.
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.

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 3-5](#) shows the aggregation and integration features and their benefits.

Table 3-5 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 get a set of tools like 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. Also, because these tools all work within the portal framework, they all 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 breaking down these silos and 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.	Easy integration with existing email, calendar, legacy, or web applications enables the portal to serve as a unified access point, enabling users—be that employees, partners, or customers—to access the information they need quickly and easily.

Understanding User Behaviors and Patterns

Study the people who will use your portal. Factors such as when they will use the portal and how they have used predecessor systems are keys to identifying your requirements. 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?
- What expectations about your portal content do your users have? How have they used predecessor web-based information or other resources that your portal will offer?

Pre-Deployment Considerations

This chapter contains the following sections:

- [Determine Your Tuning Goals](#)
- [Portal Sizing Tips](#)
- [Establish Performance Methodology](#)
- [Portal Sizing](#)
- [SRA Sizing](#)

Determine Your Tuning Goals

Before tuning your portal, work with portal system administrators and portal developers to set the portal performance objectives based upon 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 Sun Java™ System Portal Server (formerly Sun ONE 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 Sun Java System 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.

Portal Sizing Tips

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

- A business-to-consumer portal will require that you deploy SRA 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, whether it is for usual peaks after users return from a break, such as a weekend or holiday, or if it is increased usage 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 (1/4) of the overall capacity.
- In general, usage concurrencies for a business-to-employee portal are higher than a business-to-consumer portal.

Establish Performance Methodology

Once you have established your performance goals, follow the steps below to tune your portal environment.

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).

It is important to know the starting data measurement baseline so that 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 will help you to uncover the origin of performance problems.

See [“Portal Sizing” on page 61](#).

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

See [“Analysis Tools” on page 139](#)

4. Monitor customized portal applications such as portlets.

Portal Sizing

You need to establish a baseline sizing figure for your Portal Server. With a baseline figure established, you can then validate and refine that figure to account for scalability, high availability, reliability, and good performance.

The portal sizing process consists of the following steps:

1. [Establish Baseline Sizing Figures](#)
2. [Customize the Baseline Sizing Figures](#)
3. [Validate Baseline Sizing Figures](#)
4. [Refine Baseline Sizing Figures](#)
5. [Validate Your Final Figures](#)

The following sections describe these steps.

Establish Baseline Sizing Figures

Once you have identified your business and technical requirements, and mapped Portal Server features to those needs, your sizing requirements will emerge as you plan your overall Portal Server deployment. Your design decisions will help you make accurate estimates regarding Portal Server user sessions and concurrency.

NOTE Sizing requirements for a secure portal deployment using Sun Java™ System Portal Server Secure Remote Access (SRA) software are covered in [“SRA Sizing” on page 70](#).

Your Sun Java System technical representative can provide you with an automated sizing tool to calculate the estimated number of CPUs your Portal Server deployment requires. You need to gather the following metrics for input to the sizing tool:

- [Peak Numbers](#)
- [Average Time Between Page Requests](#)
- [Concurrent Users](#)
- [Average Session Time](#)
- [Search Engine Factors](#)

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 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, use these 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, in general it is not feasible to use the log to calculate the average time between requests on a user basis. 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.

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 will be registered. For example, a page containing 10 graphic files will record 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 *maximum number of concurrent users* after you calculate *maximum number of concurrent sessions*. To calculate the maximum number of concurrent users, use this formula:

$$\text{concurrent users} = \frac{\text{number of concurrent sessions}}{\text{average time between hits}}$$

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 Engine 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:

$$\text{average disk space requirement} = \frac{\text{database size}}{\text{number of RDs in database}}$$

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 they 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 [“Average Time Between Page Requests” on page 140](#).

- 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, they affect times for search result return trips.

TIP You can now give the above figures to your Sun Java System 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 lesser 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:

- 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 especially 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 are those that 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, they 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, they consequently create the need for more memory, even though the ratio of users submitting requests to Portal Server and those 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.

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](#)
- [Application Server Requirements](#)

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 will change once you begin customizing the system.

- Access to authless anonymous portal—0 ops
- Login by using the Login channel—2 BINDS, 2 SRCH
- Removing a channel from the Portal Desktop—8 SRCH, 2 MOD
- Reloading the Portal Desktop—0 ops

Application Server Requirements

One of the primary uses of Portal Server installed on an application server is to integrate portal providers with Enterprise JavaBeans™ and other J2EE™ technology stack constructs, such as JDBC and JCA, running on the application server. These other applications and modules can consume resources and affect your portal sizing.

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—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 will help 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.

NOTE Refining baseline sizing requirements for a secure portal deployment using SRA is covered in [“SRA Sizing” on page 70](#).

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 will want 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 will have:

- 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.

SRA Sizing

Use this section only if your organization is implementing a secure portal by installing SRA. As you did for portal, for SRA, you must first establish your Gateway instances baseline sizing estimate (A single machine can have one Gateway installation but multiple instances. SRA enables you to install multiple Gateways, each running multiple instances.) Your design decisions will help you make accurate estimates regarding SRA 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 SRA deployment is an iterative process. You might wish to change the inputs to generate a range of sizing results. You will then want to test these results against your original requirements. You can avoid most performance problems by formulating your requirements correctly and setting realistic expectations of SRA performance.

This section explains the following types of performance factors that the Gateway instances baseline sizing process involves:

- [Identifying Gateway Key Performance Requirements](#)
- [Advanced Gateway Settings](#)

Identifying Gateway Key Performance Requirements

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

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

NOTE Properly sizing the Gateway is difficult, and using the Gateway sizing tool is only the beginning. Gateway performance depends more on throughput than 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. See [“Secure Remote Access Example” on page 152](#) for more information.

These are the key performance factors:

- [Session Characteristics](#)
- [Netlet Usage Characteristics](#)

NOTE After you calculate these key performance factors, give the figures to your Sun Java System 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 SRA (Gateway) users

This represents the size of your user base or pool of potential users for the secure portal. See [“Concurrent Sessions” on page 139](#) for more information on estimating this number.

- Expected percentage of total users using the Gateway (at maximum load)

Apply a percentage to your total number of users to come up with this figure.

- Average time between page hits
This is how often on average a user requests a page from the portal server. See [“Average Time Between Page Requests” on page 140](#) for more information.
- Session average time
This determines how many logins per second that the Gateway must sustain for a given number of concurrent users. See [“Average Session Time” on page 141](#) for more information.

Netlet Usage Characteristics

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

- Netlet is enabled in the Sun Java™ System Identity Server (formerly Sun ONE Identity 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 come up with 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 Plugin ciphers.

Advanced Gateway Settings

Use the settings in this section to obtain more accurate results when estimating the number of Gateway instances for your deployment. These advanced Gateway settings are used as input to the automated sizing tool.

These are the advanced Gateway settings:

- [Page Configuration](#)
- [Scalability](#)
- [Sun Java System Secure Portal Pilot Measured Numbers](#)

NOTE After your Sun Java System technical representative has given you a figure for your estimated number of CPUs, consider how these related performance factors will affect this figure.

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. Because this process involves checking credentials, initializing the session, and delivering initial content, it is typically taxing on the system.

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.

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.

Secure Portal Pilot Measured Numbers

If you have numbers from a pilot of the SRA portal, 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. For example, block size determined by Netlet for a Telnet application will be different from that of FTP. It 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.

SRA Gateway and SSL Hardware Accelerators

SSL-intensive servers, such as the SRA Gateway, require large amounts of processing power to perform the encryption required for each secure transaction. Using a hardware accelerator in the Gateway speeds up the execution of cryptographic algorithms, thereby increasing the performance speed.

The Sun Crypto Accelerator 1000 board is a short PCI board that functions as a cryptographic co-processor to accelerate public key and symmetric cryptography. This product has no external interfaces. The board communicates with the host through the internal PCI bus interface. The purpose of this board is to accelerate a variety of computationally intensive cryptographic algorithms for security protocols in e-commerce applications.

See the *Portal Server Secure Remote Access 6 Administration Guide* for more information on the Sun Crypto Accelerator 1000 board and other accelerators.

NOTE The Sun Crypto Accelerator 1000 board supports only SSL handshakes and not symmetric key algorithms. This is not generic to all other cryptographic accelerators. There are other cryptographic accelerators in the market and some of them can support symmetric key encryption. See the following URL for more information:

<http://www.zeus.com/products/zws/security/hardware.html>

You could use a hardware accelerator on the Netlet Proxy and Rewriter Proxy machine and derive some performance improvement.

SRA and Sun Enterprise Midframe Line

Normally, for a production environment, you would deploy Portal Server and SRA 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 SRA in different domains on the same Sun Enterprise midframe machine. The normal CPU and memory requirements that pertain to Portal Server and SRA 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 SRA domain is in the DMZ.

Creating Your Portal Design

This chapter describes how to create your high-level and low-level portal design and provides information on creating specific sections of your design plan.

This chapter contains the following sections:

- [Portal Design Approach](#)
- [Portal Server and Scalability](#)
- [Portal Server and High Availability](#)
- [Portal Server System Communication Links](#)
- [Working with Portal Server Building Modules](#)
- [Designing Portal Use Case Scenarios](#)
- [Designing Portal Security Strategies](#)
- [Portal Server and Identity Server on Different Nodes](#)
- [Designing SRA Deployment Scenarios](#)
- [Designing for Localization](#)
- [Content and Design Implementation](#)
- [Identity and Directory Structure Design](#)

Portal Design Approach

At this point in the Portal Server deployment process, you've identified your business and technical requirements, and communicated these requirements to the stakeholders for their approval. Now you are ready to begin the design phase, in which you develop your high- and low-level designs.

Your high-level portal design communicates the architecture of the system and provides the basis for the low-level design of your solution. Further, the high-level design needs to describe a logical architecture that meets 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 Sun Java™ System Portal Server Secure Remote Access (SRA) , high availability, security (including Sun Java™ System Identity Server, and Sun Java™ System Directory Server (formerly Sun ONE Directory Server) architectural components. See [“Logical Portal Architecture” on page 79](#) for more information.

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

Once developed, the high-level design leads toward the creation of the low-level design. The low-level design specifies such items as the physical architecture, network infrastructure, Portal Desktop channel and container design and the actual hardware and software components. Once you have completed the high- and low-level designs, you can begin a trial deployment for testing within your organization.

Overview of High-Level Portal Design

The high-level design is your first iteration of an architecture approach to support both the business and technical requirements. The high-level design addresses questions such as:

- Does the proposed architecture support both the business and technical requirements?
- Can any modifications strengthen this design?
- 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 they have access to?
- Does the design account for adding more hardware to the system as required by the increase in web traffic over time?

Overview of Low-Level Portal Design

The low-level design 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 Sun Java™ System Portal Server (formerly Sun ONE 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.
- Sun Java System Identity Server 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.

Logical Portal Architecture

Your logical portal architecture defines all the components that make up the portal, including (but not limited to) the following:

- Sun Java System Portal Server itself
- Contents from RDBMs
- Third-party content providers
- Custom developed providers and content
- Integration with back-end systems such as messaging and calendaring systems
- Web container for deployment
- Role of the Content Management System
- Customer Resource Management
- Whether the portal runs in open or secure mode (requires Sun Java System 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 Sun Java System Directory Server reside here.

The logical architecture 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 up 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.

Portal Server and 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 to the Portal Server product.

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 will need. See [Chapter 4, “Pre-Deployment Considerations”](#) for more information.

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 Identity Server 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 “[Working with Portal Server Building Modules](#)” on page 87, discusses an approach to a specific type of configuration that provides optimum performance and horizontal scalability.

Portal Server and High Availability

High Availability ensures that your portal platform is accessible 24 hours a day, seven days a week. Today, organizations require that data and applications always be available. High availability has become a requirement that applies not only to mission-critical applications, but also to the whole IT infrastructure.

System availability is affected not only by computer hardware and software, but also by people and processes, which can account for up to 80 percent of system downtime. Availability can be improved through a systematic approach to system management and by using industry best practices to minimize the impact of human error.

One important issue to consider is that not all systems have the same level of availability requirements. Most applications can be categorized into the following three groups:

- **Task critical**—Affects limited number of users; not visible to customers; small impact on costs and profits
- **Business critical**—Affects significant number of users; might be visible to some customers; significant impact on costs and profits
- **Mission critical**—Affects a large number of users; visible to customers; major impact on costs and profits

The goals of these levels are to improve the following:

- Processes by reducing human error, automating procedures, and reducing planned downtime
- Hardware and software availability by eliminating single-point-of-failure configurations and balancing processing load

The more mission critical the application, the more you need to focus on availability to eliminate any single point of failure (SPOF), and resolve people and processes issues.

Even if a system is always available, instances of failure recovery might not be transparent to end users. Depending on the kind of failure, users can lose the context of their portal application, and might have to login again to get access to their Portal Desktop.

System Availability

System availability is often expressed as a percentage of the system uptime. A basic equation to calculate system availability is:

$$\text{Availability} = \text{uptime} / (\text{uptime} + \text{downtime}) * 100$$

For instance, a service level agreement uptime of four digits (99.99 percent) means that in a month the system can be unavailable for about seven hours. Furthermore, system downtime is the total time the system is not available for use. This total includes not only unplanned downtime, such as hardware failures and network outages, but also planned downtime, preventive maintenance, software upgrade, and patches.

If the system is supposed to be available seven days a week, 24 hours a day, the architecture needs to include redundancy to avoid planned and unplanned downtime to ensure high availability.

Degrees of High Availability

High availability is not just a switch that you can turn on and off. There are various degrees of high availability that refer to the ability of the system to recover from failures and ways of measuring system availability. The degree of high availability depends on your specific organization's fault tolerance requirements and ways of measuring system availability.

For example, your organization might tolerate the need to reauthenticate after a system failure, so that a request resulting in a redirection to another login screen would be considered successful. For other organizations, this might be considered a failure, even though the service is still being provided by the system.

Session failover alone is not the ultimate answer to transparent failover, because the context of a particular portal application can be lost after a failover. For example, consider the case where a user is composing a message in NetMail Lite, has attached several documents to the email, then the server fails. The user will be redirected to another server and NetMail Lite will have lost the user's session and the draft message. Other providers, which store contextual data in the current JVM™, have the same problem.

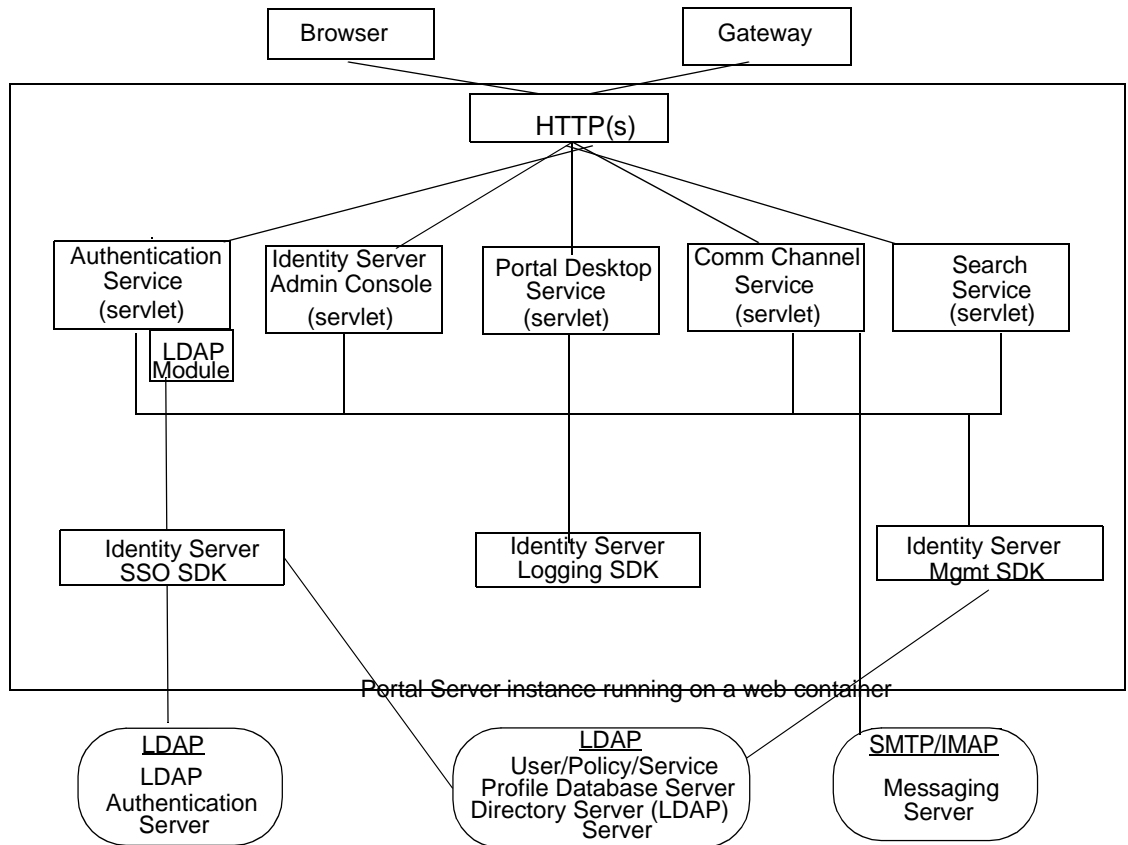
Achieving High Availability for Portal Server

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

- **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.
- **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 product (formerly Sun ONE Directory Server).)
- **Directory Server**—A number of options make the LDAP directory highly available. See [“Building Modules and High Availability Scenarios” on page 88](#) for more information.
- **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.

Portal Server System Communication Links

[Figure 5-1 on page 85](#) shows the processes and communication links of a Portal Server system that are critical to the availability of the solution.

Figure 5-1 Portal Server Communication Links

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

A user uses either a browser or the Gateway to communicate with Portal Server. This 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 Identity Server SSO SDK and the LDAP server; and between the Identity Server Management SDK and the LDAP server.

Figure 5-1 on page 85 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 Identity Server for authentication single sign-on (session) management, policy, and profile database access. Thus, Portal Server inherits all the benefits (and constraints) of Identity Server with respect to availability and fault tolerance.

By design, Identity Server's services are either stateless or they can share context data so that they can recover to the previous state in case of a service failure.

Within Portal Server, Portal Desktop and NetMail services do 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 Sun Java™ System 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.
- **SRA Gateway and Proxies**—The SRA Gateway is a standalone Java 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 those instances. Session stickiness is not required in front of a Gateway, although it is recommended for performance reasons. On the other hand, session stickiness to Portal Server instances is enforced by SRA.

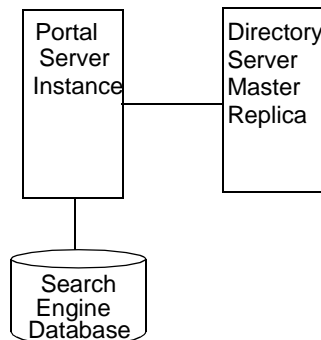
SRA includes other Java processes called Netlet Proxy and Rewriter Proxy. You use these proxies to extend the security perimeter from behind the firewall, and limit the number of holes in the DMZ. You can install these proxies on separate nodes.

Working with 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 Sun Java™ System 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 5-2](#) shows the building module architecture.

Figure 5-2 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**

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**

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**

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 even notice 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.

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 5-1 summarizes these high availability scenarios along with their supporting techniques.

Table 5-1 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

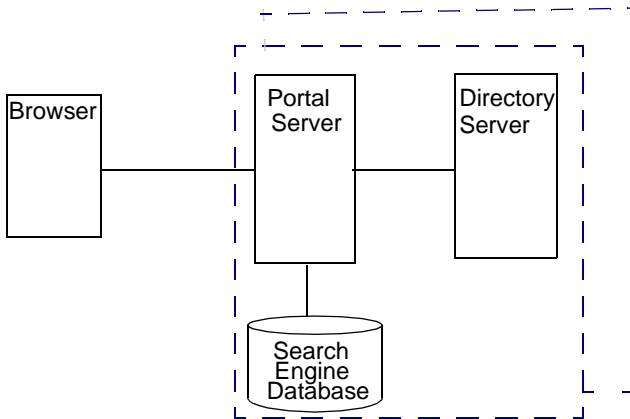
NOTE Load balancing is not provided out-of-the-box with the Sun Java System Web Server web application container.

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 5-3](#) shows a small, best effort deployment using the building module architecture.

Figure 5-3 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 Identity 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.)

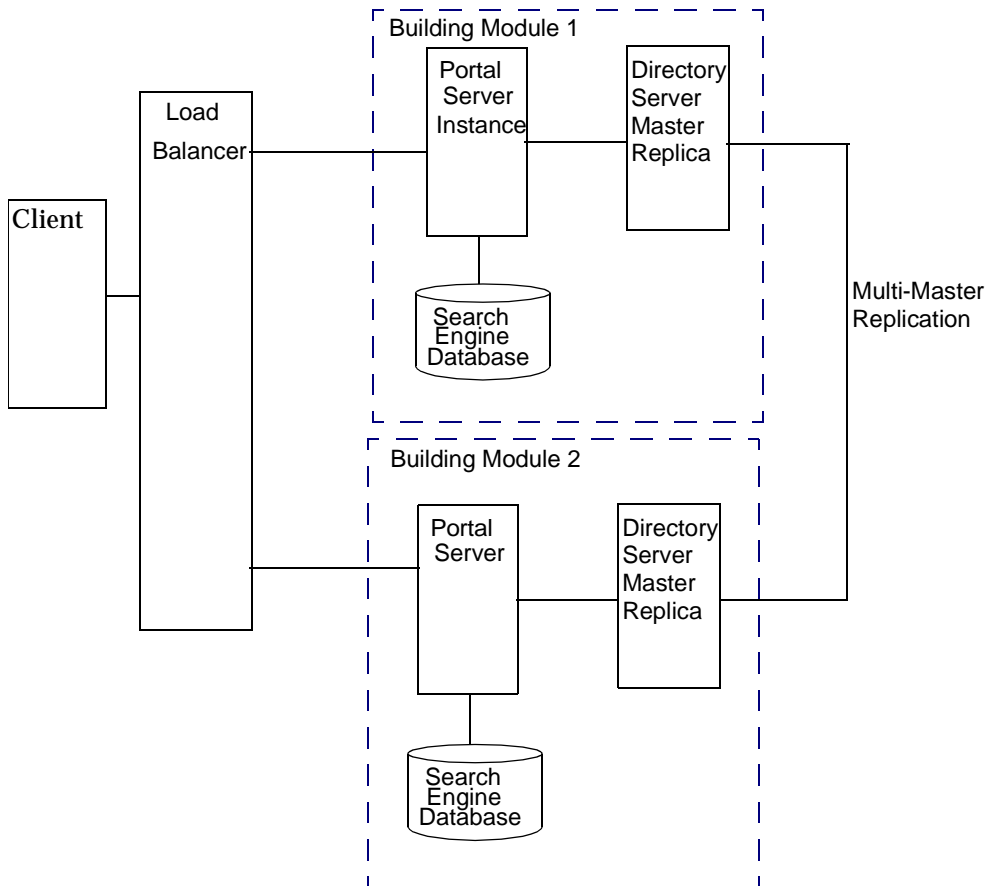
This scenario might suffice for task critical requirements. Its major weakness is that a maintenance action necessitating a system shutdown results in service interruption.

When SRA 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 5-4 No Single Point of Failure Example



As stated earlier, a building module consists of a Portal Server instance, a Directory Server master 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.

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 The Sun Java™ System Identity Server product 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 place 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 *Directory Server 6 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 Identity Server 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.

SRA follows the same replication and load balancing pattern as Portal Server to achieve NSPOF. As such, two SRA Gateways and pair of proxies are necessary in this scenario. The SRA 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 SRA Gateway performs a periodic check for availability until the first Portal Server instance is up again.

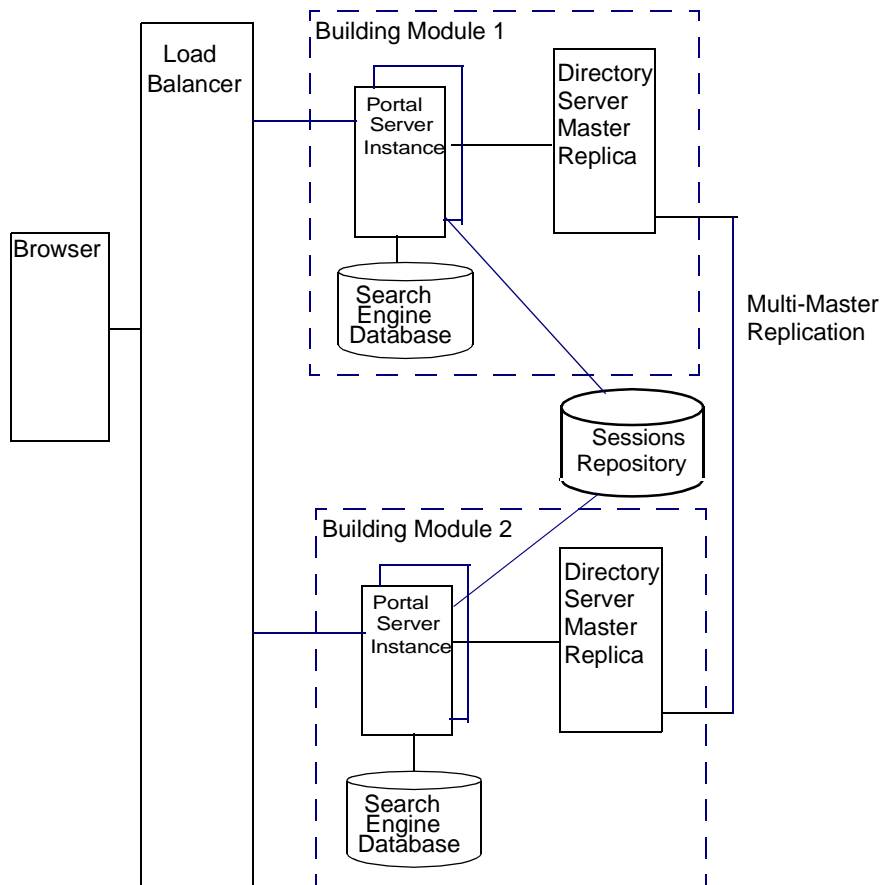
The NSPOF high availability scenario is suitable to business critical deployments. However, some high availability limitations in this scenario might not fulfill the requirements of a mission critical deployment.

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 5-5 on page 94 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 there is a crash, the application server retrieves sessions created by Building Module 1 from the sessions repository.

Figure 5-5 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. (Sun Java System Web Server and Sun Java System Application Server do not support HTTP Session failover.) See [Appendix C, “Portal Server and Application Servers”](#) 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 Constraints

The constraints on the scalability of building modules are given by the number of LDAP writes resulting from profile updates and the maximum size of the LDAP database. For more information, see [“Directory Server Requirements” on page 96](#).

NOTE If the LDAP server crashes with the `_db` files in the `/tmp` directory, most likely they will be gone 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.

Deploying Your Building Module Solution

This section describes guidelines for deploying your building module solution.

Deployment Guidelines

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 Requirements

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. (See [Figure 5-2 on page 87](#).)
- 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.

Search Engine Structure

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 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.
- The size of the Search database dictates whether more than one machine needs to host the Search database by replicating it across machines or building module. Consider using high-end disk arrays.
- Use a proxy server for caching the search hit results. When doing so, you need to disable the document level security. See the *Portal Server 6 Administration Guide* for more information on document level security.

Designing Portal Use Case Scenarios

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

Elements of Portal 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 limits 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 those 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 5-2](#) describes a use case for a portal user to authenticate with the portal.

Table 5-2 Use Case: Authenticate Portal User

Item	Description
Priority	Must have.
Context of Use	Only authenticated 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 users identify themselves only once for a complete online session. In the case that an idle timeout occurs, the users must reidentify themselves. If the portal 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 user should be advised to contact the authorized person. The identified portal users are able to access only the data and information they are authorized for.
Primary User	Portal end user.
Special Requirements	None.
Stakeholders	Portal end user.
Preconditions	The portal user is an authorized user. Standard corporate LDAP user ID. Must be provided to each employee. Authorized LDAP entry. Every employee has access to the corporate intranet. No 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 user is not yet logged in.

Table 5-2 Use Case: Authenticate Portal User *(Continued)*

Item	Description
Description	<ol style="list-style-type: none"> 1. 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. User enters previously assigned user ID and password. 5. Information is passed to Identity Server for validation. 6. If authentication passes, assign session ID and continue. 7. If authentication fails, display error message, return user to login page; decrement remaining attempts; if pre-set attempts exceed limit, notify user and lock out the account.

Designing Portal Security Strategies

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 Identity Server. In addition, you can use certificates, SSL encryption, and group and domain access.

Securing the Operating Environment

Reduce potential risk of security breaches in the operating environment by performing the following, often termed “system hardening:”

- **Minimize the size of the operating environment installation**—When installing a Sun server in an environment that is exposed to the Internet, or any untrusted network, reduce the Solaris installation to the minimum number of packages necessary to support the applications to be hosted. Achieving minimization in services, libraries, and applications helps increase security by reducing the number of subsystems that must be maintained.

The Solaris™ Security Toolkit provides a flexible and extensible mechanism to minimize, harden, and secure Solaris Operating Environment systems. The primary goal behind the development of this toolkit is to simplify and automate the process of securing Solaris systems. Please see:

<http://www.sun.com/software/security/jass/>

- **Track and monitor file system changes**—Within systems that require inclusion of security, a file change control and audit tool is indispensable as it tracks changes in files and detects possible intrusion. You can use a product such as Tripwire for Servers, or Solaris Fingerprint Database (available from SunSolve Online).

Using Platform Security

Usually you install Portal Servers in a trusted network. However, even in this secure environment, security of these servers requires special attention.

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. This raises potential security concerns as this responsibility is typically delegated to non-system administrators who can pose a threat to the system.
- **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.

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

- **Non-`root` user**—You can run Portal Server as a regular UNIX user. The security benefits of a regular user are similar to those 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 *Java Enterprise System Installation Guide* for more information.

Limiting Access Control

While the traditional security UNIX model is typically viewed as all-or-nothing, there are alternative tools that you can use, which 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 9 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, which enables a system administrator to give certain users the ability to execute a command as another user. Please see:

<http://www.courtesan.com/sudo/sudo.html>

Using a Demilitarized Zone (DMZ)

For maximum security, the Gateway is installed in the DMZ 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.

Portal Server and Identity Server on Different Nodes

Portal Server and Identity Server 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.
- Identity Server can be used by other web containers to assist with development of portal customizations.

NOTE When Portal Server and Identity Server are on different nodes, the Identity Server 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 Identity Server 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.

Login 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.

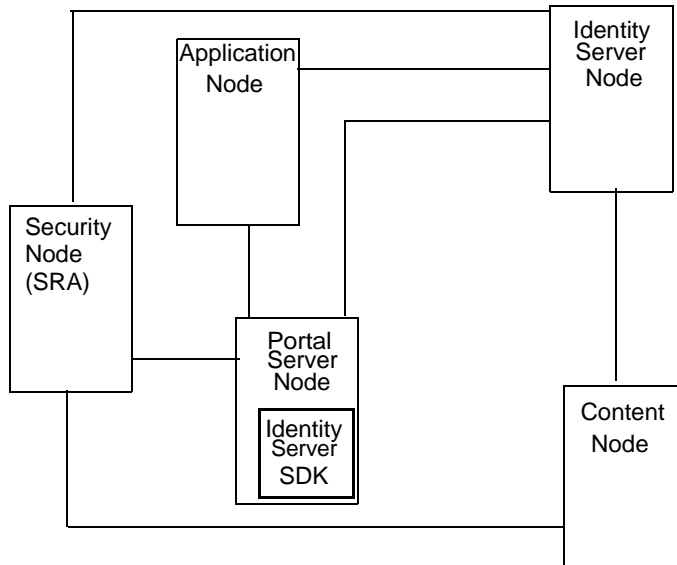
Policy API—evaluates and manages Identity Server policies and provides additional functionality for the Policy Service.

SAML API—exchanges acts of authentication, authorization decisions and attribute information.

Federation Management API—adds functionality based on the Liberty Alliance Project specifications.

Figure 5-6 illustrates Identity Server and Portal Server residing on separate nodes.

Figure 5-6 Portal Server and Identity Server on Different Nodes



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

Figure 5-7 shows two Portal Server instances configured to work with a single Identity Server and two Directory Servers where both the Identity Server and the Directory Servers operate in a Java Enterprise System Sun Clustered environment. This configuration is ideal when it is determined that the Identity and Directory Server instances are not the bottleneck.

Figure 5-7 Two Portal Servers and One Identity Server

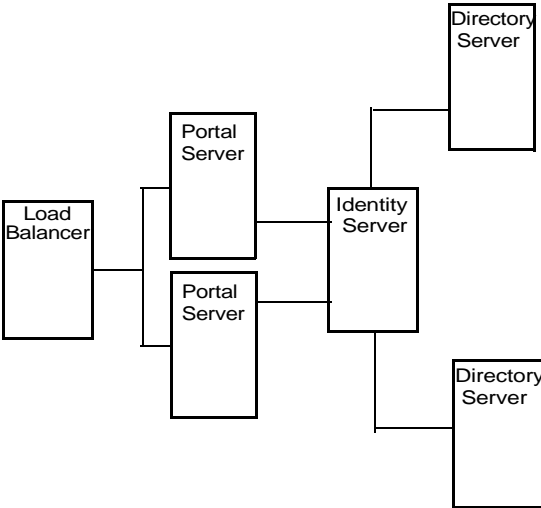


Figure 5-8 shows configuration allowing authentication throughput coming from Portal Server to be load-balanced across the two Identity Servers.

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 Identity Servers with the policy and authentication services could be on two medium-size servers.

Figure 5-8 One Portal Server and Two Identity Servers

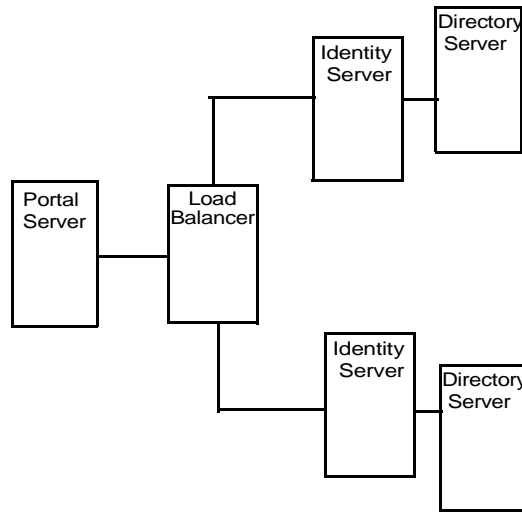


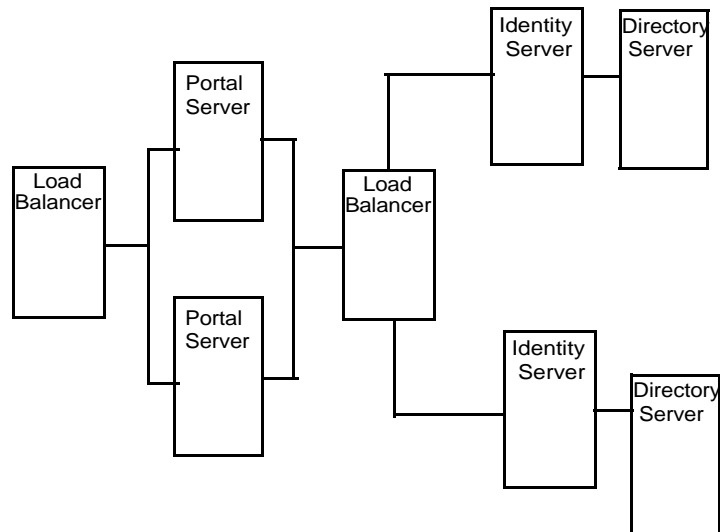
Figure 5-9 shows a configuration for maximum horizontal scalability and higher availability achieved by a horizontal server farm. Two Portal Servers can be fronted with a load balancer for maximum throughput and high availability.

Another load balancer can be put between Portal Servers and Identity Servers 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 Identity Services and Directory Services respectively. With the architecture shown in Figure 5-9 there is redundancy of services 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 Sun Java Enterprise Directory Server software schema used by the Sun Java Enterprise Identity Server 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 5-9 Two Portal Servers and Two Identity Servers



Designing SRA Deployment Scenarios

The SRA 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 SRA.

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

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.

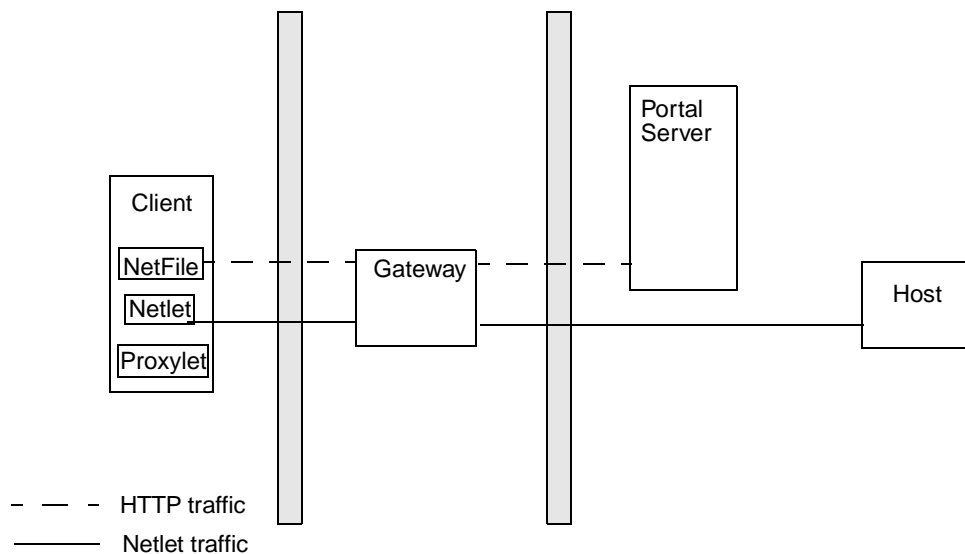
Basic SRA Configuration

Figure 5-10 shows the most simple configuration possible for SRA. 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 SRA proxies between the Gateway and the internal hosts. For Netlet traffic, the connection is direct from the Gateway to the destination host.

Without a SRA 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 5-10 Basic SRA Configuration

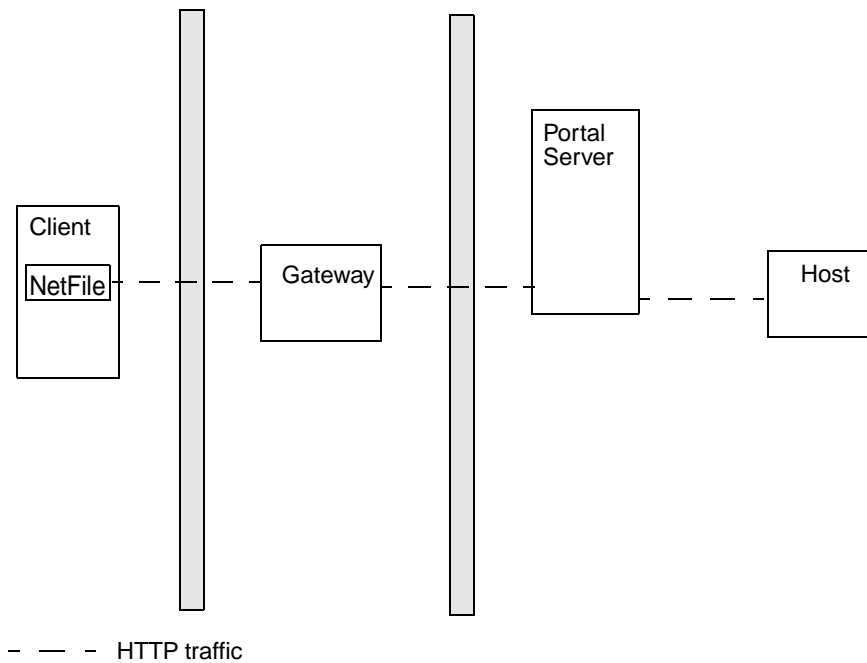


Disable Netlet

Figure 5-11 shows a scenario similar to the basic SRA 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 5-11 Disable Netlet

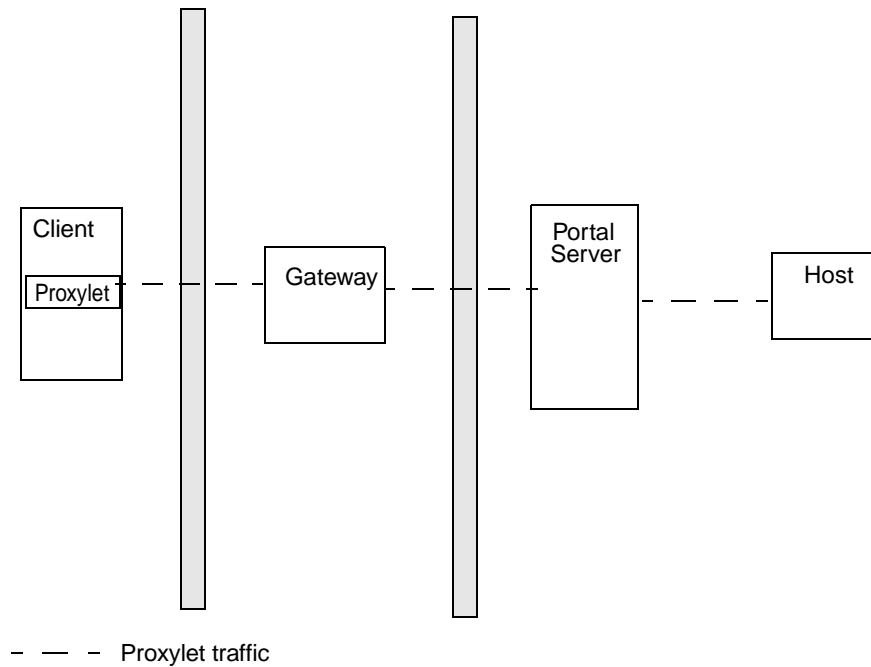


Proxylet

Figure 5-12 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.

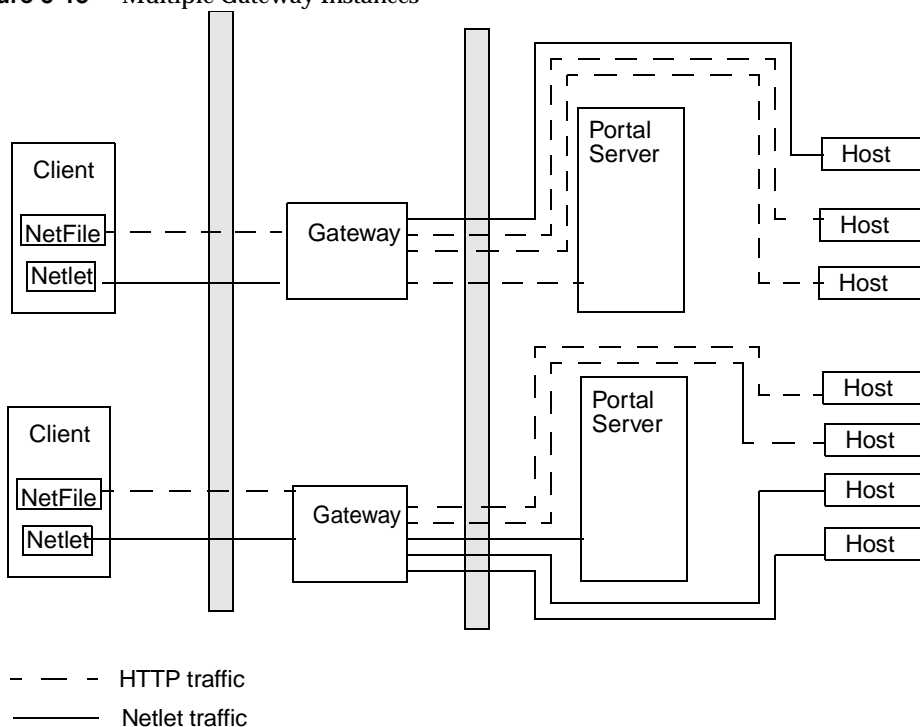
Figure 5-12 Proxylet



Multiple Gateway Instances

Figure 5-13 shows an extension of the SRA basic configuration. Multiple Gateway instances run on the same machine or multiple machines. You can start multiple Gateway instances with different profiles. See Chapter 2, “Configuring the Gateway,” in the *Portal Server Secure Remote Access 6 Administration Guide* for details.

Figure 5-13 Multiple Gateway Instances



NOTE Although Figure 5-13 on page 112 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 5-14](#) shows a configuration with a Netlet Proxy and a Rewriter Proxy on the intranet. With these proxies, only two open ports are necessary in the second firewall.

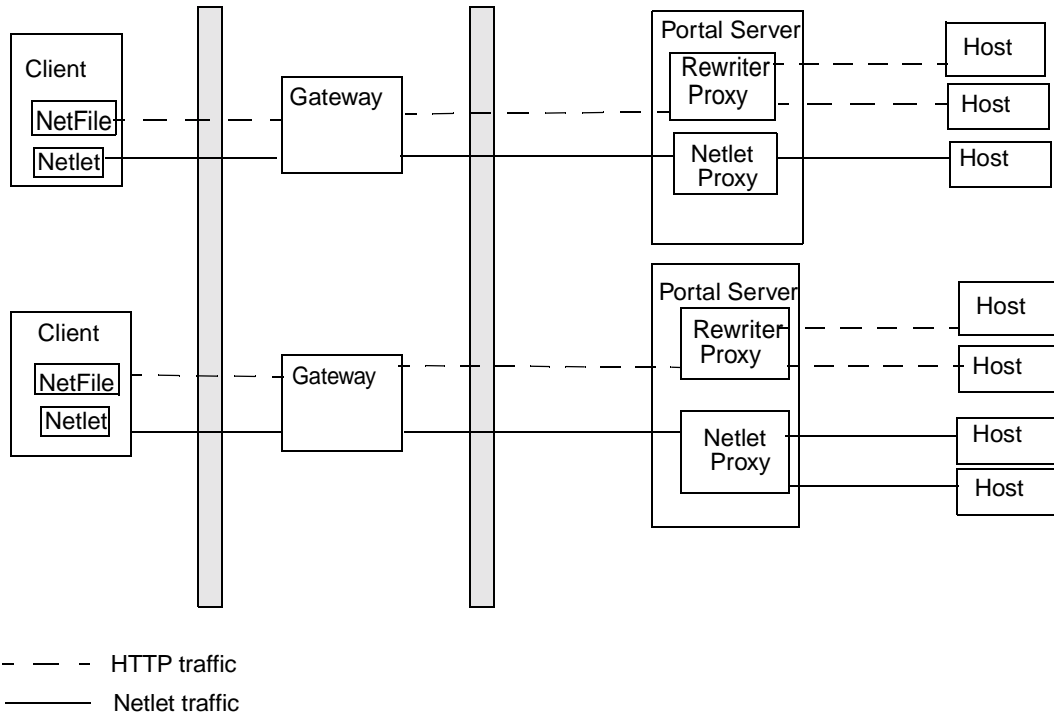
The Gateway need not contact the application hosts directly now, but will forward all Netlet traffic to the Netlet proxy and Rewriter traffic to the Rewriter Proxy. Since the Netlet Proxy is within the intranet, it can directly contact all the required application hosts without opening multiple ports 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 5-14 Netlet and Rewriter Proxies



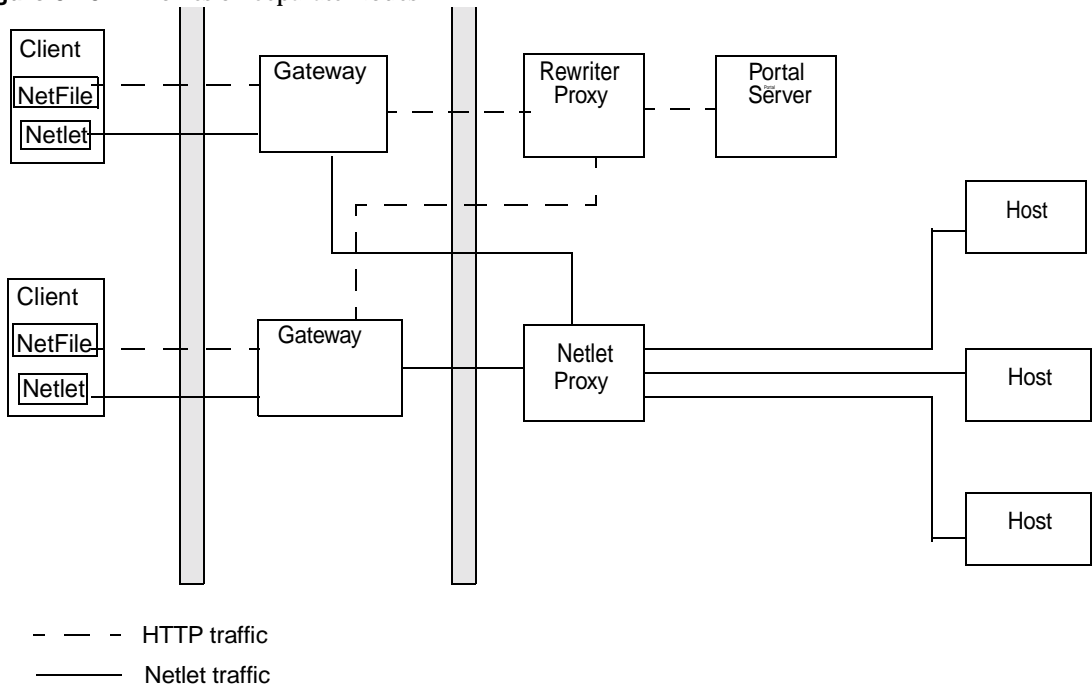
Netlet and Rewriter Proxies on Separate Nodes

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 5-15 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.

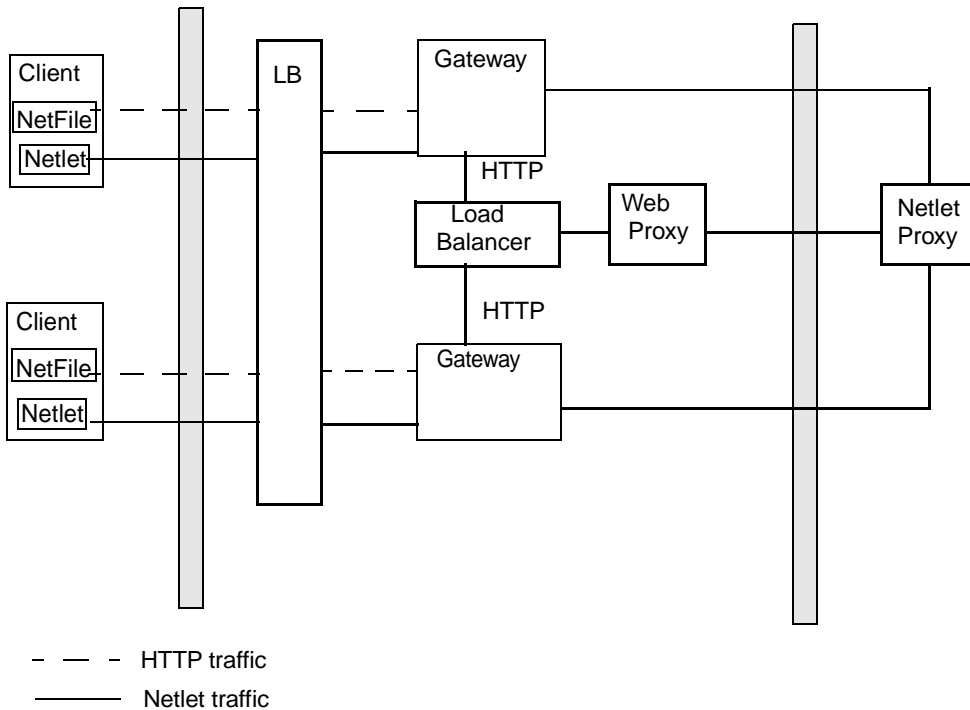
Figure 5-15 Proxies on Separate Nodes



Using Two Gateways and Netlet Proxy

Load balancers provide a failover mechanism for higher availability as there is a redundancy of services on the Portal Servers and Identity Servers.

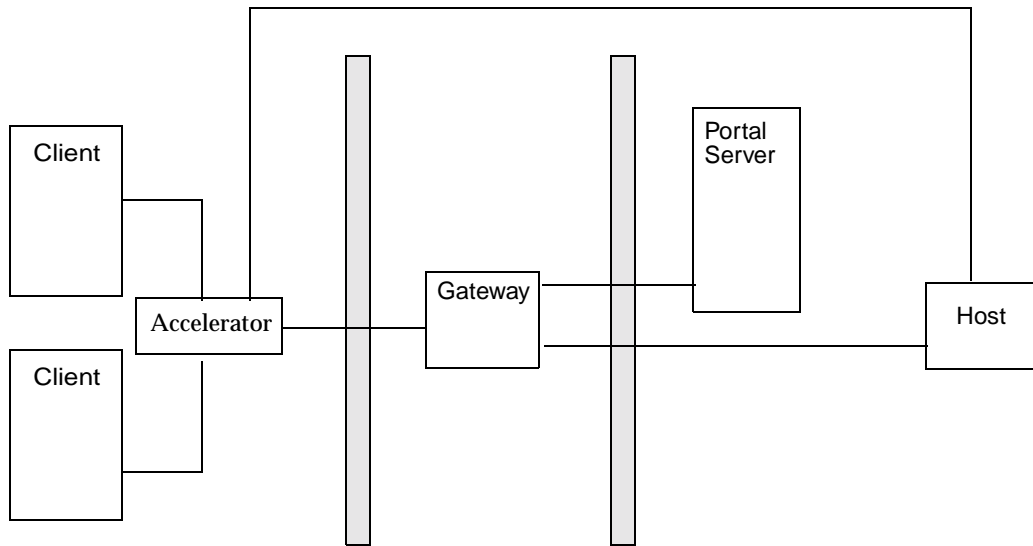
Figure 5-16 Two Gateways and Netlet Proxy



Using an 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 SRA. For information on accelerators, see the *Portal Server Secure Remote Access 6 Administration Guide*.

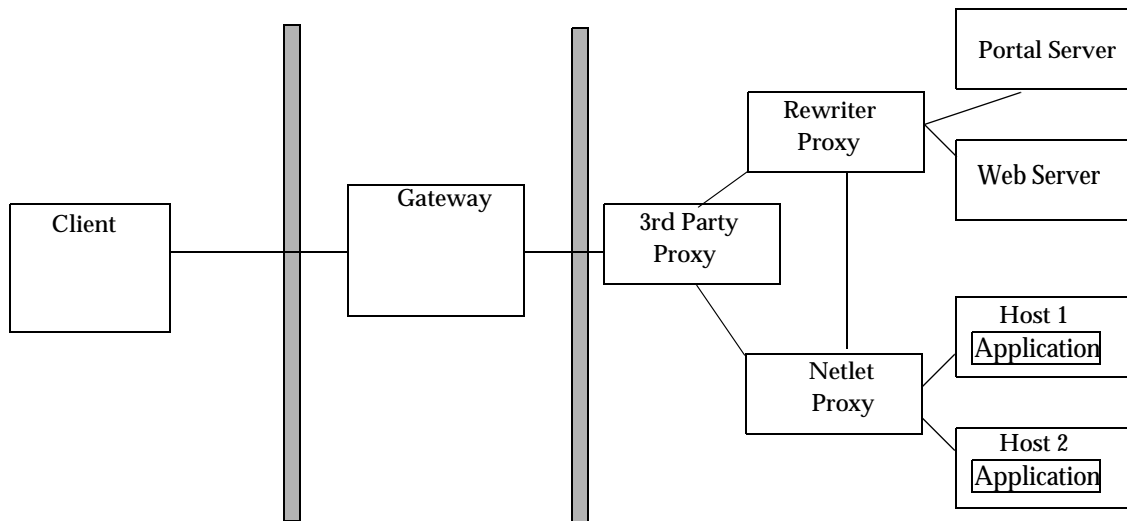
Figure 5-17 SRA Gateway with External Accelerator



Netlet with 3rd Party Proxy

Figure 5-18 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 5-18 Netlet and Third-Party Proxy

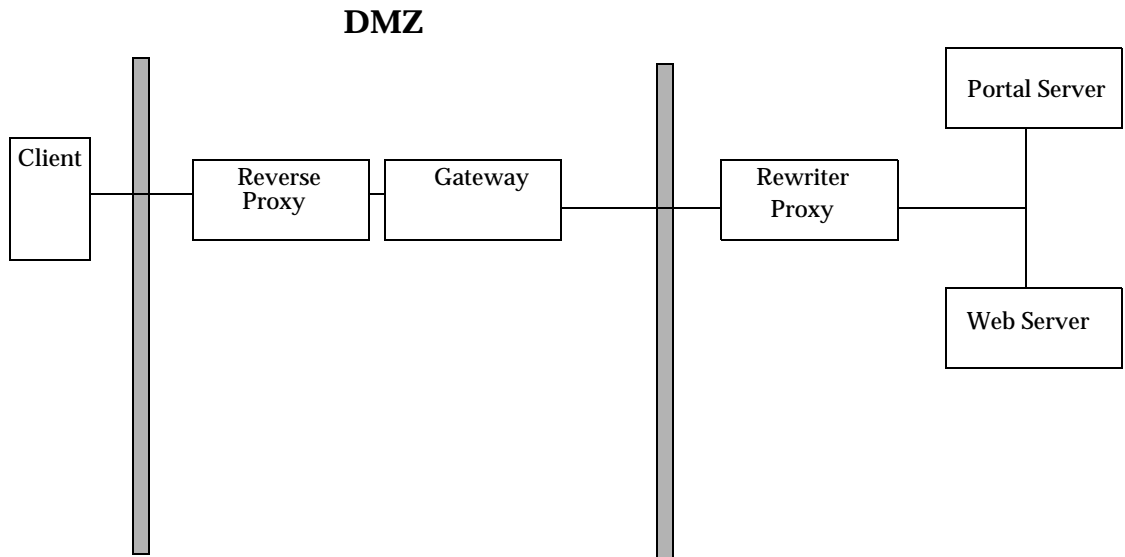


Reverse Proxy

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 5-19 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 5-19 Using a Reverse Proxy in Front of the Gateway



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:

1. Localization of the entire product into a language that we don't provide. This is usually done by a professional service organization.
2. Localization of customizable 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 *Portal Server 6 Developer's Guide* for more information on localization.

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 Identity Server 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 *Portal Server 6 Developer's Guide* and *Portal Server 6 Desktop Customization Guide* for more information.

Placement of Static Portal Content

Place your static portal content in the `web-container-install-root/SUNWam/public_html` directory or in a subdirectory under the `web-container-install-root/SUNWam/public_html` directory (the document root for the web container). Do not place your content in the `web-container-install-root/SUNWps/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.

Integration Design

This section provides information on integration areas that you need to account for in your low-level design.

Creating a Custom Identity Server Service

Service Management in Identity Server provides a mechanism for you to define, integrate, and manage groups of attributes as an Identity Server service. Readyng a service for management involves:

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

See the Identity Server documentation for more information.

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; an example is NetMail; created as an Identity Server service; accesses Portal and Identity Server APIs.
- **Third-party application**—Hosted separately from Portal Server, but accessed from Portal Server; URL Scraper, which calls Rewriter, rewrites web pages so that they can be displayed in a channel; uses Identity Server to enable single sign-on.

Independent Software Vendors

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

- **Application user interface**—This integration uses the provider API and SRA for secure access. (SRA 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 Identity Server Login API to enable portal access by using a custom authentication scheme. Examples include RSA.
- **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 Instant Messaging 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 Identity 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 Identity Server and LDAP.

NOTE Portal Server cannot currently integrate another LDAP solution. Identity Server and Portal Server rely on features not found in other LDAP implementations.

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 itself
- Application availability in secure mode (using SRA, 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.

Integrating Microsoft Exchange

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 platform optional package and is also available as part of the Java™ 2 Platform, Enterprise Edition.

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.

Identity and Directory Structure Design

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

The top of the organization tree in Identity Server 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. There is no limitation on the depth to the nested structure.

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 Identity 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.

See the *Portal Server 6 Administration Guide*, *Directory Server Deployment Guide*, and the *Identity Server Deployment Guide* for more information on planning your Identity Server and Directory Server structure.

Implementing Single Sign-On

Single sign-on (SSO) to Portal Server is managed by Identity Server. SSO provides a user with the ability to use any application that has its access policy managed by Identity Server, if allowed through the policy. The user need not re-authenticate to that application.

Various SSO scenarios include:

- **Portal web application**—The authentication comes from Identity Server, and the application validates the user credentials with Identity Server
- **Standalone web application**—The application is hosted on a separate web container, and the Identity Server Web Agent is used for authentication. This does not require application coding. Additionally, you can modify the application to validate against Identity Server directly.
- **Standalone Java application**—In this scenario, you modify the application to validate user credentials against Identity Server directly.
- **Non-Identity Server aware application**—In this scenario an application stores a user's credentials and provides them as needed. However, this is not an ideal SSO solution, as the user needs to re-authenticate if credentials change.

Portal Desktop Design

The performance of Portal Server itself 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 Aggregation Strategy

The options for implementing portal channels for speed and scalability include:

- 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 what kind of data it is. 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 it 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:

- `URLScrapperProvider`—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 double-digit milliseconds or hundredths of milliseconds 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.

`URLScrapperProvider` 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 `URLScrapperProvider` needs to rewrite that picture to present it to portal users.

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

To use `URLScrapperProvider` 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 is able to retrieve content. On the first fetch of content, performance for this provider is usually in the low teen 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 code. 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 Identity Server 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 `URLScrapperProvider`. This provider uses the JAXP 1.2 JAR files provided by Sun Java™ System Web Server software.
- **LDAP-based provider**—This type of provider retrieves information about a user and use of personalization from user profile. It stays efficient as long as the number of LDAP attributes stored is low. In general, this type of provider is a good performer, second only to the file scraper provider within `URLScrapperProvider`.
- **Database provider**—This type of provider utilizes a back-end database for its content. It requires that you build database connection pooling and that you use small queries (either single queries, or no more than a couple). You might also have to perform extra work for HTML formatting. In general, this type of provider is the worst performer, due to its use of database connection pooling, large database queries, poor coding, or lack of indexing on the retrieved data. Additionally, once the data has been retrieved, the portal needs to perform a large amount of processing to display the data in the Portal Desktop. If you use this type of provider, push as much data processing logic to the database as possible. Also, benchmark your portal performance with and without database channels in the user profile.

Client Support

Portal Server supports the following browsers as clients:

- Internet Explorer 5.5 and 6.0
- Netscape™ Communicator 4.7x or higher

See the *Portal Server 6 Release Notes* for updates to this list.

Multiple client types, whether based on HTML, WML, or other protocols, can access Identity Server and hence Portal Server. For this functionality to work, Identity Server 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, Identity Server defines client data only for supported HTML client browsers, including Internet Explorer and Netscape Communicator. See the Identity Server documentation for more information.

Sun Java™ System Portal Server Mobile Access 6.3 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.

The Production Environment

This chapter describes how to monitor and tune Sun Java™ System Portal Server software, including the Sun Java™ System Portal Server Secure Remote Access product.

This chapter contains the following sections:

- [Moving to a Production Environment](#)
- [Monitoring Portal Server](#)

Moving to a Production Environment

Moving to a production environment occurs after you have thoroughly tested your portal and operated it as a trial deployment to test and refine your design.

Monitoring and Tuning

Monitoring and tuning your portal deployment is an ongoing, cyclical process, in which you look for bottlenecks and other performance issues.

With monitoring and tuning your portal, keep the following points in mind:

- Beginning with the trial portal, define a baseline performance for your deployment, before you add in the full complexity of the project.
- 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 will be the first to max out as you increase the activity to the portal. This will indicate the amount of headroom you have as well as identify where to expend your energies.
- Measure and monitor your traffic regularly to verify your model.
- Use the model for long-range scenario planning. Understand how dramatically you will have to change your deployment to meet your overall growth projections for upcoming years.
- In a production system, 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.
- Run the `perftune` script on one of your production servers to know if the thread limits are being reached and if you need to further tune web server parameters.

The `perftune` script (located in the `portal-server-install-root/SUNWps/bin` directory), bundled with Sun Java System Portal Server, automates most of the tuning process discussed in this guide.

See the *Portal Server 6 Administration Guide* for information on the `perftune` script.

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, when time and money are usually running short, to begin this documentation phase. Documenting your portal should occur as an ongoing activity throughout the entire deployment.

Monitoring Portal Server

This section describes the variables that affect portal performance, as well as the portal monitoring you can perform. Areas to monitor include:

- Sun Java™ System Identity Server
- Portal Desktop
- Sun Java™ System Directory Server
- Java™ Virtual Machine

While there are emerging technologies that will enable you to perform detailed monitoring of Portal Server services, this section focuses on the basic but extensive set of hardware and software issues that determine the overall performance of a portal deployment.

Specifically, portal performance is determined by the throughput and latency of which it is capable over a period of time. You must conduct a baseline performance analysis as soon as possible. The baseline performance analysis confirms that your portal substantially conforms to published performance numbers. Establishing a performance baseline helps you to understand infrastructure issues that can severely impact the performance of a production portal.

Nevertheless, when maintaining a properly performing portal, you must look at a broad set of issues. The following sections explain those issues in terms of portal performance variables and provides guidelines for determining portal efficiency.

NOTE These rules also apply for performance, scalability, and stress tests.

Memory Consumption and Garbage Collection

Before reading this section, read the following document on tuning garbage collection with the Java Virtual Machine, version 1.4.2:

<http://java.sun.com/docs/hotspot/gc1.4.2/index.html>

Portal Server requires substantial amounts of memory to provide the highest possible throughput. In fact, the *portal-server-install-root/SUNWps/bin/perftune* tuning script sets the heap size maximum to 2 GB. This size is divided between the new generation, which receives 256 MB (one eighth of the space) and the old generation, which receives the rest. At initialization, a maximum address space is virtually reserved but does not allocate physical memory unless it is needed. The complete address space reserved for object memory can be divided into the young and old generations.

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.

Even with a large heap size, after a portal instance has been running under moderate load for a few days, most of the heap will appear to be used because of the lazy nature of the GC. The GC will start performing full garbage collections until the resident set size (RSS) reaches approximately 85 percent of the total heap space. Those garbage collections can have a measurable impact on performance.

For example, on a 900 MHz UltraSPARCIITM, 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 and it is important to understand the impact on performance and the frequency of full GCs. In production, full GCs will go unnoticed most of the time, but any monitoring scripts that measure the performance of the system need to account for the possibility that a full GC might occur.

Measuring the frequency of full GCs is sometimes the only way to determine if the system has a memory leak. It is important to 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 JVMTM` parameter.

CPU Utilization

When deployed using the building module concept (as described in [Chapter 5, “Creating Your Portal Design”](#)), Portal Server has a capable, scalable CPU architecture that also degrades gracefully under high loads.

However, when monitoring a production site, track CPU utilization over time. Load usually comes in spikes and keeping ahead of those 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 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.

Identity Server Cache and Sessions

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

You can enable the `amSSO` and `amSDKStats` logs to monitor the number of active sessions on the server and the efficiency of the Sun Java System Directory Server cache. These logs are located by default in the `/var/opt/SUNWam/debug` 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 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 will be 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 not include a built-in reporting mechanism to monitor portal usage information by portal users. This includes which channels are accessed, how long they are accessed, and the ability to build a user behavioral pattern of the portal. However, it is relatively simple to build a simple Java™ servlet that would intercept every Portal Server Desktop request, extract the SSO token, save the user access information to a log, then redirect the user to the intended URL. Such a construct would be based on custom attribute extensions to the Identity Server schema.

Installed Product Layout

This appendix describes the Sun Java™ System Portal Server (formerly Sun ONE Portal Server) directory structure and properties files used to store configuration and operational data.

Directories Installed for Portal Server

[Table A-1](#) shows the platform-specific directory structures that are installed for Sun Java System Portal Server.

Table A-1 Portal Server Directories

Description	Location
Default installation directory	<i>portal-server-install-root</i> /SUNWps
Default installation directory for configuration information	/etc/ <i>portal-server-install-root</i> /SUNWps
Default installation directory for SDK	<i>portal-server-install-root</i> /SUNWps/sdk
Temporary files	/usr/tmp
Debug files	/var/ <i>portal-server-install-root</i> /SUNWam/debug
Log files	/var/ <i>portal-server-install-root</i> /SUNWam/log /var/ <i>portal-server-install-root</i> /SUNWps/ <i>instance-directory</i>
Search Engine logging, configuration, and data directories	/var/ <i>portal-server-install-root</i> /SUNWps/ <i>instance-directory</i> / <i>log-directory</i>
Container and channel display profile	<i>portal-server-install-root</i> /SUNWps/samples/desktop/dp-org.xml
Provider display profiles	<i>portal-server-install-root</i> /SUNWps/samples/desktop/dp-providers.xml

Table A-1 Portal Server Directories (Continued)

Description	Location
HTML template files	<i>/etc/portal-server-install-root/SUNWps/desktop/default/channelname.template</i>
JSP template files	<i>/etc/portal-server-install-root/SUNWps/desktop/default/JSPchannelname</i>
Command-line utilities	<i>portal-server-install-root/SUNWps/bin/</i>
Tag library definitions	<i>/etc/portal-server-install-root/SUNWps/desktop/default/tld/*.tld</i>
Display profile DTD	<i>portal-server-install-root/SUNWps/dtd/psdp.dtd</i>
Java properties files	<i>portal-server-install-root/SUNWam/locale</i>

Directories Installed for SRA

This section describes the Sun Java™ System Secure Remote Access (SRA) directory structure and configuration files used to store configuration and operational data.

[Table A-2](#) shows the platform-specific directory structures that are installed for Sun Java System Secure Remote Access.

Table A-2 Portal Server, SRA Directories

Description	Location
Default installation directory	<i>portal-server-install-root/</i>
Default installation directory for ISun Java™ System Identity Server executables, the web server, and the deployed applications	<i>portal-server-install-root/SUNWam</i>
Default installation directory for configuration information	<i>/etc/portal-server-install-root/SUNWps</i>
Log files	<i>/var/portal-server-install-root/SUNWam/logs</i>
Debug log files	<i>/var/portal-server-install-root/SUNWps/debug</i>

Configuration Files

All Portal Server and SRA configuration data is stored using the Sun Java System Identity Server Services Management function. Identity Server provides the bootstrap configuration file that is needed to find the Sun Java™ System Directory Server.

The `platform.conf` file contains the details that the Gateway needs. By default, the `platform.conf` file is located at:

```
/etc/opt/SUNWps
```


Analysis Tools

The Sun Java Enterprise System and SDK include default setting options to ensure a satisfactory out-of-the-box experience. However these options might not provide optimal performance for your web applications in the Sun Java™ System Portal Server (formerly Sun ONE Portal Server) production environment. This section describes some alternative options and basic tuning techniques.

NOTE The tuning settings discussed in this section focus on Sun Java System Portal Server residing on the Solaris platform. However, the principles can be applied to other generic Unix type operating systems.

[Table B-1](#) below lists the performance analysis tools that will help in providing feedback for tuning the Portal Server and its web container. In addition to performance issues, many of these tools can be used to detect other types of bottlenecks at the overall operating system level.

Many tool descriptions provide sample output, suggestions for interpreting output results, tips on improving output results, and links to related sites.

Table B-1 Performance Analysis Tools

Category	Type	Name	Parameters	Usage
Analysis Tool	Solaris 8 and Solaris 9	mpstat		CPU utilization
		iostat		Disk I/O subsystem
		netstat		Network subsystem
			-I hme) 10	Interface bandwidth
			-sP tcp	TCP kernel module

Table B-1 Performance Analysis Tools

Category	Type	Name	Parameters	Usage
			<code>-a grep hostname wc -1</code>	Socket connection count
	Portal Server on App Server container	<code>verbose:gc</code>		Garbage collection
Tuning Parameters	Solaris 8 and Solaris 9	<code>/etc/system</code>	Various	Performance
		<code>/etc/rc2.d/ttuning parameters file</code>	Various	TCP kernel tuning parameters

mpstat

The `mpstat` utility is a useful tool to monitor CPU utilization, especially with multithreaded applications running on multiprocessor machines, which is a typical configuration for enterprise solutions.

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.

Output

```
#mpstat 10
```

```

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

Note the much higher `intr` and `ithr` values for certain CPUs. Solaris will select some CPUs to handle the system interrupts. The CPUs and the number that are chosen depend on the I/O devices attached to the system, the physical location of those devices, and whether interrupts have been disabled on a CPU (`psradmin` command).

- `intr`- interrupts
- `intr`- thread interrupts (not including the clock interrupts)
 - `csw`- Voluntary Context switches. When this number slowly increases, and the application is not IO bound, it may indicate a mutex contention.
 - `icsw`- Involuntary Context switches. When this number increases past 500, the system is under a heavy load.
 - `smtx`- if `smtx` increases sharply, for instance from 50 to 500, it is a sign of a system resource bottleneck (ex., network or disk).
 - `usr`, `sys` and `idl` - Together, all three columns represent CPU saturation. A well-tuned application under full load (0% idle) should be within 80% to 90% `usr`, and 20% to 10% `sys` times, respectively. A smaller percentage value for `sys` reflects more time for user code and less preemption, which result in greater throughput for Portal application.

Considerations

It is important to make available to your application as many CPUs as it can efficiently use. As an example, you get the best performance from one instance from 2 CPUs, it is reasonable to expect that creating 14 2CPU processor sets would yield the best performance.

An increasing `csw` value shows an increase with network use. A common cause for a high `csw` value is the result of having created too many socket connections--either by not pooling connections or by handling new connections inefficiently. If this is the case you would also see a high TCP connection count when executing `netstat -a | wc -l`. Please refer to the `netstat` section.

Do you observe increasing `icsw`? A common cause of this is preemption, most likely because of an end of time slice on the CPU.

iostat

The `iostat` tool gives statistics on the disk I/O subsystem. The `iostat` command has many options. More information can be found in the man pages. The following typical options provide information on locating I/O bottlenecks.

Output

```
#iostat -xn 10
```

```

                                extended device statistics
  r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
  0.0    0.0     0.0     0.0  0.0  0.0   0.0   0.0   0   0 fd0
  2.7   58.2   14.6 2507.0  0.0  1.4   0.0  23.0   0  52 d0
 47.3    0.0 2465.6    0.0  0.0  0.4   0.0   8.8   0  30 d1

```

What to Look For

- `%b` - Percentage of time the disk is busy (transactions in progress). Average `%b` values over 25 could be a bottleneck.
- `%w` - Percentage of time there are transactions waiting for service (queue non-empty).
- `asvc_t`- Reports on average response time of active transactions, in milliseconds. It is mislabeled `asvc_t`; it is the time between a user process issuing a read and the read completing. Consistent values over 30ms could indicate a bottleneck.

Considerations

Add more disks to the file system. When using a single disk file system, consider, upgrading to a hardware or software RAID is the next logical step. Hardware RAID is significantly faster than software RAID and is highly recommended. A software RAID solution would add additional CPU load to the system.

Depending on storage hardware and application behavior, there may be a better block size to use besides the ufs default of 8192k. Please consult Solaris System Administration Guide.

netstat

The `netstat` tool gives statistics on the network subsystem. It can be used to analyze many aspects of the network subsystem, two of which are the TCP/IP kernel module and the interface bandwidth. An overview of both uses follow.

`netstat -I hme0 10`

These `netstat` options are used to analyze interface bandwidth. The upper bound (max) of the current throughput can be calculated from the output. The upper bound is reported because the `netstat` output reports the metric of packets, which don't necessarily have to be their maximum size. The upper bound of the bandwidth can be calculated using the following equation:

Bandwidth Used = (Total number of Packets) / (Polling Interval (10)) * MTU (1500 default).

The current MTU for an interface can be found with: `ifconfig -a`

```
netstat -I hme0 10 Output
#netstat -I hme0 10
  input   hme0      output           input (Total)    output
 packets errs  packets errs  colls  packets errs  packets errs  colls
122004816 272  159722061 0    0    348585818 2582  440541305 2    2
  0        0    0        0    0    84144    0    107695    0    0
  0        0    0        0    0    96144    0    123734    0    0
  0        0    0        0    0    89373    0    114906    0    0
  0        0    0        0    0    84568    0    108759    0    0
  0        0    0        0    0    84720    0    108800    0    0
```

What to Look For

- `colls`- collisions. If your network is not switched, then a low level of collisions is expected. As the network becomes increasingly saturated, collision will increase and eventually will become a bottleneck. The best solution for collisions is a switched network.

- `errs`- errors. The presence of errors could indicate device errors. If your network is switched, errors indicate that you are nearly consuming the bandwidth capacity of your network. The solution to this problem is to give the system more bandwidth, which can be achieved through more network interfaces or a network bandwidth upgrade. This is highly dependent on your particular network architecture.

Considerations

- If network saturation is occurring quickly (saturation at less than 8CPUs for an application server running on a 100mbit Ethernet), then an investigation to ensure conservative network usage is a good first step.
- Increase network bandwidth. Steps that possibly can be taken: upgrade to a switched network, more network interfaces are a possible solution or upgrade to a higher bandwidth network to accommodate your network traffic demand.
`netstat -sP tcp`

These `netstat` options are used to analyze the TCP kernel module. Many of the fields reported represent fields in the kernel module that indicate bottlenecks. These bottlenecks can be addressed using the `nnd` command and the tuning parameters referenced in the `/etc/inet`

netstat -sP tcp Output

```
#netstat -sP tcp
```

```
TCP      tcpRtoAlgorithm      =      4      tcpRtoMin      =      400
```

<snip>

```
tcpInDupSegs      =      1144      tcpInDupBytes      =132520
```

```
tcpInPartDupSegs  =        1      tcpInPartDupBytes  =      416
```

```
tcpInPastWinSegs  =        0      tcpInPastWinBytes  =        0
```

```
tcpInWinProbe     =       46      tcpInWinUpdate     =       48
```

```
tcpInClosed       =      251      tcpRttNoUpdate     =      344
```

```
tcpRttUpdate      =1105386      tcpTimRetrans      =      989
```

```
tcpTimRetransDrop =        5      tcpTimKeepalive    =      818
```

```
tcpTimKeepaliveProbe=    183      tcpTimKeepaliveDrop =        0
```



```

tcpListenDrop      =      0      tcpListenDropQ0    =      0
tcpHalfOpenDrop    =      0      tcpOutSackRetrans  =     56

```

What to look for

- `tcpListenDrop`- If after several looks at the command output the `tcpListenDrop` continues to increase, it could indicate a problem with queue size.

Considerations:

- A possible cause of increasing `tcpListenDrop` is the application throughput being bottlenecked by the number of executing threads. At this point increasing application threads may be a good thing to try.
- Increase queue size. Increase the request queue sizes using `ndd`. More information on other `ndd` commands referenced in the *Solaris Administration Guide*.

```
ondd -set /dev/tcp tcp_conn_req_max_q <value>
```

```
ondd -set /dev/tcp tcp_conn_req_max_q0 <value>
```

```
netstat -a | grep <your_hostname> | wc -l
```

Running this command gives a rough count of socket connections on the system. There is a limit on how many connections can be open at one time; therefore, it is a good tool to use when looking for bottlenecks.

```
netstat -a | grep <your_hostname> | wc -l Output
```

```
#netstat -a | wc -l
```

```
34567
```

What to Look For

- `socket count` - If the number returned is greater than 20,000 then the number of socket connections could be a possible bottleneck.

Consider the following:

- Decrease the point where number of anonymous socket connections start.

```
ondd -set /dev/tcp tcp_smallest_anon_port <value>
```

- Decrease the time a TCP connection stays in `TIME_WAIT`.

```
ondd -set /dev/tcp tcp_time_wait_interval <value>
```

Tuning Parameters for /etc/system

Table B-2 is a list of /etc/system tuning parameters used during the performance study. The changes are applied by appending each to the /etc/system file.

Table B-2 /etc/system Options

/etc/system Option	Description
set rlim_fd_max=<value>	"Hard" limit on file descriptors that a single process might have open. To override this limit requires superuser privilege.
set tcp:tcp_conn_hash_size=<value>	Controls the hash table size in the TCP module for all TCP connections. Along with tune_t_flushr, autoup controls the amount of memory examined for dirty pages in each invocation and frequency of file system sync operations.
set autoup=<value>	The value of autoup is also used to control whether a buffer is written out from the free list. Buffers marked with the B_DELWRI flag (file content pages that have changed) are written out whenever the buffer has been on the list for longer than autoup seconds. Increasing the value of autoup keeps the buffers around for a longer time in memory.
set tune_t_fsflushr=<value>	Specifies the number of seconds between fsflush invocations.
set rechoose_interval=<value>	Number of clock ticks before a process is deemed to have lost all affinity for the last CPU it ran on. After this interval expires, any CPU is considered a candidate for scheduling a thread. This parameter is relevant only for threads in the timesharing class. Real-time threads are scheduled on the first available CPU.

A description of all /etc/system parameters can be found in the *Solaris Tunable Parameters Reference Manual*.

Table B-3 is a list of TCP kernel tuning parameters. These are known TCP tuning parameters that affect most performance on Portal Servers. Recommended values for these parameters are discussed in the *Identity Server Customization and API Guide*.

Table B-3 TCP/IP Options

TCP/IP Options	Description
ndd -set /dev/tcp tcp_xmit_hiwat 65535	The default send window size in bytes. The default receive window size in bytes.
ndd -set /dev/tcp tcp_rcv_hiwat 65535	

Table B-3 TCP/IP Options

TCP/IP Options	Description
<code>ndd -set /dev/tcp tcp_cwnd_max 65535</code>	The maximum value of TCP congestion window (cwnd) in bytes.
<code>ndd -set /dev/tcp tcp_rexmit_interval_min 3000</code>	The default minimum retransmission timeout (RTO) value in milliseconds. The calculated RTO for all TCP connections cannot be lower than this value.
<code>ndd -set /dev/tcp tcp_rexmit_interval_ max 10000</code>	The default maximum retransmission timeout value (RTO) in milliseconds. The calculated RTO for all TCP connections cannot exceed this value.
<code>ndd -set /dev/tcp tcp_rexmit_interval_ initial 3000</code>	The default initial retransmission timeout value (RTO) in milliseconds
<code>ndd -set /dev/tcp tcp_time_wait_interv al 60000</code>	The time in milliseconds a TCP connection stays in TIME-WAIT state. Refer to RFC 1122, 4.2.2.13 for more information.
<code>ndd -set /dev/tcp tcp_keepalive_interv al 900000</code>	The time in milliseconds a TCP connection stays in KEEP-ALIVE state. Refer to RFC 1122, 4.2.2.13 for more information.
<code>ndd -set /dev/tcp tcp_conn_req_max_q <value></code>	The default maximum number of pending TCP connections for a TCP listener waiting to be accepted by <code>accept(SOCKET)</code> .
<code>ndd -set /dev/tcp tcp_conn_req_max_q0 <value></code>	The default maximum number of incomplete (three-way handshake not yet finished) pending TCP connections for a TCP listener.
<code>ndd -set /dev/tcp tcp_ip_abort_interva l <value></code>	Refer to RFC 793 for more information on TCP three-way handshake.
<code>ndd -set /dev/tcp tcp_ip_abort_interva l <value></code>	The default total retransmission timeout value for a TCP connection in milliseconds. For a given TCP connection, if TCP has been re-transmitting for <code>tcp_ip_abort_interval</code> period and it has not received any acknowledgment from the other endpoint during this period, TCP closes this connection.

Tuning Parameters for `/etc/system`

Portal Server and Application Servers

This appendix provides an overview of the Sun Java™ System Portal Server (formerly Sun ONE Portal Server) product and its support for application servers.

This appendix contains the following sections:

- [Introduction to Application Server Support in Portal Server](#)
- [Portal Server on an Application Server Cluster](#)

Introduction to Application Server Support in Portal Server

The Sun Java System Portal Server product provides support for the following application servers to be used as the web application container, in addition to the Sun Java™ System Web Server software:

- Sun Java™ System Application Server 7.0 Update 3
- BEA WebLogic Server™ Server 8.1 SP 2
- IBM WebSphere® Application Server 5.1

NOTE Portal Server runs in the context of a web application container, which can be either a web server (Sun Java™ System Web Server product) or one of the application servers mentioned above, depending on your deployment. This chapter assumes that the web application container is an application server.

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™ and other J2EE™ technologies that run in the application server container
- Use application server clustering, which provides scalability and high availability (currently only available on BEA WebLogic Server™ and IBM WebSphere® Application Server)
- Use session failover in clustering (currently available on BEA WebLogic Server™ and Sun Java™ System Application Server)

Portal Server on an Application Server Cluster

This section describes how Sun Java™ System Application Server 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 Sun Java™ System Identity Server are not pure web applications. Instead, they are composed of local files residing on a machine and three web applications: portal, amserver, and amconsole. These three web applications run in a web application container, which runs in an application server web application container.

The Java Enterprise System installs and configures the local files, configures the local application server, then deploys the three WAR files on the local web application container. The WAR files themselves are not self-contained; they 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:

1. Install Portal Server on all machines using the same configuration settings.

2. Deploy the three web applications (portal, amserver, and amconsole) to the cluster.

The following sections explain what it means to enable Portal Server to run on an application server cluster.

Overview of Sun Java System Application Server

Sun Java System Application Server 7, Standard Edition Server does not support server clustering or session failover. However, it does support enhanced web tier support by enabling you to partition HTTP and HTTPS traffic arriving on the same web server instance to multiple application servers in the middle tier. This facility in Standard Edition can be used to partition traffic to different application servers from the web server tier by using the provided reverse proxy plugin.

While Platform Edition is limited to a single application server instance (that is, a single JVM™ process) per administrative domain, you can configure the Standard Edition with multiple application server instances per administrative domain.

In addition, the Enterprise Edition supports multi-tiered, multi-machine, clustered application server deployments.

See the following Sun Java System Application Server documentation for more information:

http://docs.sun.com/db/coll/s1_asse_en

Overview of BEA WebLogic Server Clusters

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/docs61/cluster/index.html>

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/docs61/cluster/servlet.html#1009453>

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 *Java Enterprise System 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`.

Overview of IBM WebSphere Application Server

The IBM WebSphere Application Server product uses the following definitions:

- **Administrative domain**—The logical space in which the configurations for various objects in the WebSphere environment reside. Inside one administrative domain you start with an application server. This is the default installation.
- **Server group**—A server group is a template for creating additional, nearly identical copies of an application server configuration. (This is the equivalent of BEA's cluster.)
- **Clones**—A copy of the server group, on the same machine or on different machines. Clones are the equivalent of BEA's managed servers.

See the IBM WebSphere Application Server documentation for more information:

<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/welcome.html>

WebSphere Advanced Server 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.

Troubleshooting Your Portal Deployment

This appendix describes how to troubleshoot the Sun Java™ System Portal Server software and the Sun Java™ System Portal Server Secure Remote Access (SRA) software.

This appendix contains the following sections:

- [Troubleshooting Portal Server](#)
- [Troubleshooting SRA](#)

Troubleshooting Portal Server

This sections contains troubleshooting information for Sun Java System Portal Server.

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:

```
/ns-slapd -D /usr/ldap/slapd-server -i /usr/ldap/slapd-server/logs/pid
```

Sun Java™ System Identity Server:

```
identity-server-install-root/SUNWam/bin/doUnix -c 8946
```

Sun Java™ System Portal Server:

```
./uxwdog -d portal-server-install-root/SUNWam/servers/https-server/config
```

```
ns-httpd -d portal-server-install-root/SUNWam/servers/https-server/config
```

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
```

Log Files

Examine the following log files for errors.

Web Server (errors and access):

```
web-container-install-root/SUNWam/servers/https-server/logs
```

Sun Java System Directory Server:

```
/var/opt/SUNWam/logs
```

Recovering the Search Database

The Search database maintains recoverable transaction logs. Thus, under normal circumstances, you do not have to do anything to recover the database. Recovery from errors and transient conditions such as a full disk are straight forward. If desired, maintain Search database archives and restore from an archive in case you lost the entire database. In this scenario, you would copy the archive to the original database to recover it.

► To Recover the Database

1. Stop all processes accessing the database, including the Portal Server instance.
2. Use the `rdmgr -R` command to recover.

Working with the Display Profile

If you need to troubleshoot the XML contents of your portal's display profile, it is useful to extract it 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 `dpadmin` command to extract the display profile. For example:

```
./dpadmin list -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password
-d "o=sesta.com,o=isp" > /tmp/displayxml
```

This example puts the contents of the display profile into the `/tmp/displayxml` file.

► **To Reload the Display Profile**

1. Login as administrator.
2. Use the `dpadmin` command to reload the display profile. For example:

```
./dpadmin modify -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password
-d "o=sesta.com,o=isp" /tmp/updated_displayxml
```

This example reloads the contents of the display profile from the `/tmp/updated_displayxml` file.

High CPU Utilization for Portal Server Instance

When using the Cisco Content Services Switch, you might see a very high CPU utilization on the Portal Server instance with Sun Java™ System Web Server error file showing the following message every five seconds.

```
[20/Jan/2003:16:53:36] failure ( 5926): Error accepting connection -5928, oserr=130
(Connect aborted)
```

The cause of this error is a “sticky bit” setting within the Cisco Content Services Switch that is causing these errors. These load balancers periodically ping the servers (every five seconds) to verify that they are alive. After turning off the “sticky bit” setting, which disables the ping to the server every 5 seconds, the errors will no longer show up in Sun Java System Web Server.

Configuring a Sun Java System Portal Server Instance to Use 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:

```
java.net.UnknownHostException: services.xmethods.net
```

► To Configure Usage of an HTTP Proxy for a Portal Server Instance

1. Change directories to the portal server install root directory containing the configuration for the instance.

```
cd portal-server-install-root/SUNWam/servers/https-servername/config
```

2. Edit the `server.xml` file within this directory and add the following lines:

```
http.proxyHost=proxy-host
```

```
http.proxyPort=proxy-port
```

```
http.nonProxyHosts=portal-host
```

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.

Troubleshooting SRA

This section describes how to capture information that Sun Java System support personnel need to troubleshoot problems in your deployment.

Debugging the Gateway

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:

```
gateway-install-root/SUNWam/config/AMConfig-instance-name.properties
```

2. Set the debug level:

```
com.iplanet.services.debug.level=
```

The debug levels are:

error - Only serious errors are logged in the debug file. Rewriter usually stops functioning when such errors occur.

warning - Warning messages are logged.

message - All debug messages are logged.

off - No debug messages are logged.

3. Specify the directory for the debug files in the following property of the `AMConfig-instance-name.properties` file:

```
com.iplanet.services.debug.directory=/var/opt/SUNWam/debug
```

where `/var/opt/SUNWam/debug` is the default debug directory.

4. Restart the Gateway from a terminal window:

```
gateway-install-root/SUNWps/bin/gateway -n gateway-profile-name start
```

Introduction to shooter

The `shooter` tool captures all the information that the development and support team will require to troubleshoot problems in your deployment of the Sun Java™ System Portal Server Secure Remote Access product. You can also run this tool on a Portal Server machine.

This tool captures the following data:

- Installation type—Determines if the installation has Sun Java System Portal Server with Sun Java System Secure Remote Access core, or Portal Server with SRA
- System configuration related information—Determines the host, domain, operating system, version, CPU type and speed, clock speed, and memory available
- Processors, processor sets, and the SRA processes bound to them
- SRA installation log
- The `platform.conf` file(s)

- The settings in the Gateway script such as the JVM™ settings including heap usage, and library path
- Gateway service settings
- Tuning settings in various files used for configuring Sun Java System Identity Server, Sun Java System Directory Server, and Sun Java™ System Web Server
- Output of the Java™ garbage collection
- A memory or process footprint while the Gateway was being used
- Formatted debug log files
- Rewriter rulesets

NOTE This tool collects information only for the instance of the Gateway that you specified during installation.

Using shooter

The `shooter` tool includes five files as described below.

`shooter.sh`

This is the main script. Run this script after a test or just before starting a test on the SRA installation.

From `portal-server-install-root/bin/perf`, type:

```
./shooter.sh
```

This tool collects data under a temporary folder and displays the folder name.

`gctool.pl`

This script collects and formats the garbage collection output from the JVM.

To run `gctool`, start the Gateway, and type the following to redirect the output to this script and allow collection throughout the test.

```
/etc/init.d/Gateway -n default start | gctool.pl
```

NOTE Before running `gctool`, ensure that you include `-verbose:gc` in the Gateway script in the “CMD” section. The Gateway script resembles the following:

```
-server -verbose:gc -Xms1G -Xmx2G
-XX:+OverrideDefaultLibthread -XX:ThreadStackSize=128
-XX:MaxPermSize=128M -XX:PermSize=128M -XX:MaxNewSize=256M
-XX:NewSize=256M
```

At the end of the test period, run `shooter` to collect the output of `gctool` along with other data.

memfoot.sh

This script tracks the memory footprint of a process. Start this script after starting the Gateway and allow it to run during the duration of the test. The largest process with the given name or PID is tracked after every specified number of seconds.

To run `memfoot`, type:

```
./memfoot java 60
```

The output of this script is a time-stamped process status file. The `shooter` tool collects this output along with the rest of the data.

uniq.pl

This script is used internally by `shooter` to find unique lines and their count. The advantage over the system `uniq` script is that it finds non-adjacent unique lines.

GWDump.class

This class is called internally by `shooter` to obtain the Gateway settings in the Identity Server administration console.

SRA Log Files

Examine the following log files for errors.

Gateway:

```
/var/opt/SUNWps/debug/srapGateway_Gateway-hostname_Gateway-profile-name
```

NetFile:

`/var/opt/SUNWps/debug/srapNetFile`

Netlet:

`/var/opt/SUNWps/debug/srapNetlet_Gateway-hostname_Gateway-profile-name`

Portal Deployment Worksheets

This appendix provides worksheets to help with the portal deployment process.

This appendix contains the following sections:

- [Portal Assessment Worksheets](#)
- [Portal Key Design Task List](#)

Portal Assessment Worksheets

Use these worksheets to learn more about your organization's business needs and potential areas of concern around deploying portals.

Table E-1 General Questions

1. 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
 - Eliminating the cost of maintaining many point solutions
 - Expanding the reach of the customer base for your services
 - Reducing the time to deploy new business 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
 - To comply with governmental regulations
 - Personalizing the user experience
 - Needing to gather business intelligence on the usage of services
-

Table E-1 General Questions

2. How many portals does your organization already have?
 3. What types are they (business-to-employee, business-to-consumer, business-to-business, ISP)?
 4. If you have more than one, do you have a need to reduce the number? Integrate? Federate?
 5. Do you have departmental portals?
 6. What is the extent of your Web presence? How many web sites do you have?
 7. List the top ten application services of value to you, that you would like to expose by using Portal Server to your partners? Suppliers? Customers? Employees?
 8. Who is the target community for your portal?
-

Table E-2 Organizational Questions

1. Who are the stakeholders of this portal?
 2. 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?
 3. Would an application service exposed by using the portal be made up of smaller business applications managed by an inter-departmental business process?
 4. Who would “own” this portal (the infrastructure)?
 5. Who would own the content?
 6. How do you plan to recruit additional business owners within your organization to contribute their content or applications for your portal?
 7. What project management, architect, and technical implementation resources do you have available to help develop this portal?
 8. Who sets the policies for web site characteristics such as look and feel and presentation?
-

Table E-3 Business Service-level Expectations Questions

1. Are your development projects consistent? Do you manage their risk?
 2. How does your development team work with your test, deployment, and operations groups?
 3. How many different platforms does your organization currently support?
 4. How secure is your information? How consistent is the security?
 5. Are these challenges getting better, or getting worse?
 6. How do you plan to recruit additional business owners within your organization to contribute their content or applications for your portal?
 7. What project management, architect, and technical implementation resources do you have available to help develop this portal?
 8. Who sets the policies for web site characteristics such as look and feel and presentation?
-

Table E-4 Content Management Questions

1. Do you have a content or document management system?
 2. Do you have any defined workflow to manage the development and publication of content?
 3. Do you have a taxonomy defined?
 4. How well is your information tagged and categorized?
 5. How is your enterprise content developed, managed, tracked, and published?
 6. Do you have a need for syndicated content on your portal? If so, what?
 7. What proportion of your content is dynamic versus static?
-

Table E-5 User Management and Security Questions

1. How would you segment, categorize, and relate (hierarchically) your user community?
 2. What are your current and future security policies?
 3. Do various departments own or maintain their private view of the customer?
 4. Do you have an enterprise directory?
-

Table E-6 Business Intelligence Questions

1. Do you have a need to gather, store, analyze, and provide information for enterprise decision-making?
 2. Do you already employ any data analysis or OLAP tools?
 3. At what level(s) do you need to collect business intelligence (enterprise-wide, division, department, project, onetime event)?
-

Table E-7 Architecture Questions

1. Do you already have an existing architecture strategy?
 - 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?
 2. Are there organizational issues that are hindering a successful implementation of a new IT architecture?
 3. For the top ten services that you would like deployed by using a portal, what platform and architecture do you need to support?
 4. How do these services authenticate users and manage access control
 5. How do you programmatically gain access to these services?
 6. What is your current and future messaging (email) and collaboration architecture?
 7. What is your current and future enterprise directory architecture?
 8. What technologies are used for application integration?
-

Table E-7 Architecture Questions *(Continued)*

-
9. What is the size of the target user community?
 10. How many concurrent users?
 11. What is the range of portal usage?
 12. What is the geographical distribution of your user base?
 13. Do you currently have or have a future need for non-Web access (Wireless, Voice/IVR)?
 14. Would your customer base require internationalization of content and services?
 15. What server platform technologies do you use?
 16. What development environments, tools do you use?
 17. What development methodologies do you employ?
-

Portal Key Design Task List

Table E-8 lists the major portal deployment phases and key 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 E-8 Key Design Task List *(1 of 7)*

Major Phases and Tasks	Subtasks
<i>1. Project Start and Coordination</i>	
Project Planning	<ul style="list-style-type: none"> • Perform general project management

Table E-8 Key Design Task List (2 of 7)

Major Phases and Tasks	Subtasks
Project Plan Review	<ul style="list-style-type: none"> • Review pre-implementation • Review business requirements • Review technical requirements • Review architectural documents • Review hardware and infrastructure
Coordinate Resources	<ul style="list-style-type: none"> • Identify skills required • Identify resources • Schedule resources • Assemble project team members • Review work plan with project team members
Define Requirements	<ul style="list-style-type: none"> • Collect business requirements • Summarize requirements • Confirm functional requirements • Collect technical requirements • Summarize technical requirements • Confirm technical requirements • Prepare combined requirements document • Deliver requirements
<i>2. Design</i>	
Develop Solution Architecture	<ul style="list-style-type: none"> • Design software architecture • Design server topology • Document architecture
Develop Portal Integration	<ul style="list-style-type: none"> • Understand system integration approach • Define container and channel layout • Define content aggregation • Define SSO approach • Develop custom Netlet and authentication modules
User Interface Design	<ul style="list-style-type: none"> • Prepare or modify user interface design • Develop or update screen specifications • Review and approve user interface model

Table E-8 Key Design Task List (3 of 7)

Major Phases and Tasks	Subtasks
Directory Design	<ul style="list-style-type: none"> • Design organizations, suborganizations, roles, and users • Define privileges • Review shared data requirements • Establish data transfer protocols • Create temporary or intermediate tables • Test temporary or intermediate tables • Document design approach • Deliver design document • Obtain appropriate stakeholder and organizational consensus
<i>3. Develop and Integrate</i>	
Install Software for Testing and Development Environments	<ul style="list-style-type: none"> • Install Sun Java System Portal Server software and optionally Sun Java™ System Portal Server, Secure Remote Access software (install appropriate supporting software) • Install application server, if needed • Install other software • Configure server software • Test server software components • Document test findings
Install Server Software for Development Environment	<ul style="list-style-type: none"> • Install Portal Server and optionally Sun Java™ System Portal Server Secure Remote Access (formerly Sun ONE Portal Server, Secure Remote Access) • Install application server, if needed • Install other software • Test server software components • Document test findings
Software Configuration	<ul style="list-style-type: none"> • Apply specific software configuration requirements • Create product configuration matrix

Table E-8 Key Design Task List (4 of 7)

Major Phases and Tasks	Subtasks
Sun Java™ System Portal Server, (formerly Sun ONE Portal Server) Sun Java™ System Application Server, and Other Software Modifications	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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)	<ul style="list-style-type: none"> • Perform integration • Prepare package integration test plan • Perform integration test • Produce package integration test results

Table E-8 Key Design Task List (5 of 7)

Major Phases and Tasks	Subtasks
Reporting	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Establish test plan
Plan User Acceptance Test	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 E-8 Key Design Task List (6 of 7)

Major Phases and Tasks	Subtasks
Conduct Integration and System Test	<ul style="list-style-type: none"> • 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
<i>4. Deployment Production</i>	
Confirm Approach	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Review existing documentation of results of tests • Validate scope, objectives, and critical success factors • Update deployment approach • Review and approve deployment
Implement Deployment	<ul style="list-style-type: none"> • 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 E-8 Key Design Task List (7 of 7)

Major Phases and Tasks	Subtasks
Training	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Create “run book” for system administrators

Portal Key Design Task List

Glossary

Refer to the Java Enterprise System Glossary (<http://docs.sun.com/doc/816-6873>) for a complete list of terms that are used in this documentation set.

SYMBOLS

/etc/opt/SUNWps directory 135
/etc/system tuning parameters 146
/opt/SUNWps directory 135
/opt/SUNWps/sdk directory 135

A

accelerators
 and Gateway 39, 74
access control
 Gateway 39
 limiting 102
 NetFile 44
 Netlet 41
Access Control Instructions 124
administration console tasks 26
aggregation
 description and benefits 57
 strategy 125
Allowed URLs and Denied URLs lists
 Gateway 39
 NetFile 44
amSDKStats log 133
amSSO log 133
analysis tools 139
anonymous Desktop 127
applets, NetFile 43
application servers

 clustering 150
 requirements 68
 support for 149
applications
 degree of integration 123
 dynamic port 40
 integrating 121
 portal 122
 static port 40
 third-party 122
authentication 26, 52, 53, 127
 and LDAP 85
 basic authentication 108
 custom 122
 Gateway 37
 Identity Server Web Agent 125
 modes 38
 PDC 38
 Portal Server 26
 UNIX 43
authentication server 86
average session time 64
average time between page requests 63

B

back-end servers 66
banner 80
baseline portal performance analysis 129
basic authentication 37

- BEA WebLogic 151
- bottlenecks
 - and building modules 96
 - and tuning 129
- building modules 87
 - and Directory Server 92
 - and high availability 88
 - and Search Engine 96
 - and transparent failover 94
 - constraints 95
 - deploying 95
 - description 87
- business objectives 49
- business requirements 49
- business-to-consumer portal 60
- business-to-employee portal 60

C

- cache hit ratio 133, 134
- caching appliance, and reverse proxy 80
- channels
 - description 121
 - organizing content 120
- checkpointing mechanisms 89
- chroot environment 36
- Citrix 50
- client detection API 128
- client support 128
- clustering
 - application servers 150
 - session failover 150
- communication links 84
- components
 - Identity Server 26
 - NetFile 42
 - Portal Server 26
 - SRA 35
- concurrent sessions 62, 64
- concurrent users 63
- configuration data 30
- configuration files

- Portal Server and SRA 137
- configuring, HTTP proxy 158
- content, placing 121
- CPU utilization 133
 - and mpstat utility 140
 - high with Cisco Content Services Switch 157
- CPUs
 - and Gateway instances 74
 - and vertical scaling 81
 - estimating number 62, 73
- credentials, NetFile 43
- customizing
 - affects on performance 67
 - baseline figures 67
 - Identity Server service 121

D

- data centers, and sizing 69
- Database provider 127
- delegated administration 53
- Demilitarized Zone, description 80
- deployment
 - bottlenecks 96
 - building modules 95
 - building modules and guidelines 95
 - ISP hosting 31
 - providers 126
 - requirements 49
 - software 29
- deployment scenarios 90
 - and SRA 90
 - building modules 90
 - no single point of failure 91
 - SRA 108–119
 - transparent failover 94
- designing
 - for integration 121
 - for localization 120
 - security strategies 100
 - SRA deployment scenarios 108–119
 - use case scenarios 97
- Desktop type 73

- directories
 - installed for Portal Server 135
 - installed for SRA 136
- Directory Information Tree 124
- directory replica 92
- Directory Server
 - and building modules 92
 - clustering 89
 - description 27
 - requirements 96
 - structure design 124
- Directory service
 - description 52
- directory structure
 - SRA 136
- Discussion channel 55
- display profile 120
 - and JSP files 127
 - and troubleshooting 156
 - DTD location 136
 - extracting 157
 - location for provider 135
 - properties 120
 - reloading 157
- DIT 124
- DMZ, description 80, 102
- document level security 97
- documenting the portal 130
- dpadmin command 157
- dp-org.xml file 135
- dp-providers.xml file 135
- dynamic port applications 40
- dynamic web applications 29

E

- encryption 100
 - 128-bit 108
 - 40-bit 38
 - Netlet 39
 - Portal Server 26
 - symmetric key encryption 75
- Enterprise JavaBeans 68

- error logging level 130
- example use case 99
- extracting the display profile 157

F

- failover 84, 89
- fault tolerance, and high availability 83
- file compression, NetFile 45
- FTP, NetFile 42

G

- Gateway
 - accelerators 39
 - access control 39
 - advanced settings 73
 - Allowed URLs and Denied URLs 39
 - and HTTP basic authentication 37
 - and Non Authenticated URL 39
 - and proxies 86
 - authentication 37
 - chroot environment 36
 - description 25
 - high availability 84
 - HTTP and HTTPS 36
 - logging 39
 - multihomed 36
 - multiple instances 36
 - Netlet traffic 38
 - overview 36
 - page configuration 73
 - performance requirements 71
 - profile 37
 - proxies 37
 - session
 - information, Gateway 38
 - session stickiness 37
 - SSL 38
 - SSL hardware accelerators 74
 - Gateway profile 37
 - gctool.pl tool 160

H

hardware redundancy 89, 90
 heap size 132
 high availability 82
 and building modules 88
 and Portal Server components 83
 degrees of 83
 high-level architecture, typical installation 31
 high-level portal design, overview 78
 horizontal scaling, description 81
 HTTP and HTTPS modes, and Gateway 36
 HTTP basic authentication 37
 HTTP proxy, configuring 158
 HttpSession failover 89

I

IBM WebSphere Application Server, overview 153
 identifying requirements 49
 Identity management, features and benefits 52
 Identity Server
 administration console 26
 cache and sessions 133
 components 26
 customizing 121
 description 52
 description and benefits 53
 organization tree 124
 single sign-on 26
 Identity Server SDK, components 103
 Identity Server Web Agent 125
 implementing, single sign-on 125
 independent software vendors, types 122
 installing, as a regular user 102
 integrating applications 121
 integration design 121
 interface bandwidth, and netstat 143
 Internet Explorer 128
 iostat tool 142
 ISP hosting deployment 31
 isp organization 124

ISVs, types 122

J

Java compatibility 30
 Java properties files 136
 JavaScript
 in Rewriter 46
 Portal Server Desktop 80
 JavaServer Pages 127
 JAXP 27
 JCA, and sizing 68
 jCIFS, NetFile 42
 JDBC, and sizing 68
 JSP template files, location 136
 JSPPProvider 127
 JSPPProvider 120
 JSS 27

L

LDAP
 authentication 85
 transaction numbers 68
 LDAP-based provider 127
 LDIF file 121
 legacy servers 28
 load balancing
 and high availability 89
 and Portal Server failures 92
 and Rewriter 47
 and SRA 93
 with SRA 84
 locale file 136
 localization 120
 log files
 and troubleshooting 156
 location 135
 SRA 161
 logging

- errors 130
- Gateway 39
- number of active sessions 133
- login type 73
- LoginProvider 127
- low-level portal design, overview 79

M

- memfoot .sh script 161
- Microsoft Exchange 40
 - and Netlet 40
 - integrating 123
 - Netlet Proxy 42
- MIME types,NetFile 45
- monitoring
 - active sessions 133
 - Portal Server 129
- moving to a production environment 129
- mpstat 140
- multihomed Gateway 36
- multi-master
 - and Directory Server 88
 - configuration 89, 96
- multiple network connections, Portal Server 41
- multithreading
 - and mpstat 140
 - NetFile 45

N

- NetFile
 - access control 44
 - Allowed URLs or Denied URLs 44
 - applet 43
 - components 42
 - compression 45
 - compression types 45
 - initialization 43
 - multithreading 45
 - overview 42

- Portal Server Desktop 43
- search 45
- security 44
- validating credentials 43
- Netlet
 - access control 41
 - and Microsoft Exchange 40
 - and third -party applications 41
 - application integration 41
 - encryption 72
 - encryption (ciphers) 39
 - overview 39
 - requests and Gateway 36
 - split tunneling 41
 - traffic 38
 - usage characteristics 72
- Netlet Proxy
 - and software crash 84
 - and transparent failover 95
 - Microsoft Exchange 42
 - overview 42
 - third party proxy 42
- NetMail 122
- NetMail Lite 83
- Netscape Communicator 128
- netstat tool 143
- NFS, NetFile 42, 43
- Non Authenticated URL list, and Gateway 39
- Novell domain 43

O

- open mode 23
- Outlook client 40

P

- packaging 29
- pcAnywhere 50
- PDC authentication 38
- peak numbers 62

- performance
 - analysis tools 139
 - baseline analysis 131
 - building modules 95
 - CPU utilization 133
 - establishing methodology 60
 - garbage collection 132
 - Identity Server cache and sessions 133
 - memory consumption 132
 - TCP kernel 146
 - thread usage 134
 - tuning parameters 146
 - perftune script 130
 - personalization
 - description and benefits 56
 - retrieval 127
 - placement of portal content 121
 - platform security 101
 - Portal Desktop
 - configuration 65
 - design 125
 - portal key design task list 167
 - Portal Server
 - and high availability 83
 - and Identity Server on different nodes 103
 - and load balancers 92
 - building modules 87
 - client support 128
 - communication links 84
 - components 26
 - configuration files 135
 - design approach 77
 - directory structure 135
 - documenting functions 130
 - hardware and applications 66
 - high availability 82
 - high-level design 78
 - instance and servlets 85
 - instance description 86
 - logical architecture 79
 - low-level design 79
 - mapping features to needs 52
 - multiple network connections 41
 - nodes 27, 36
 - open mode 23
 - overview 22
 - scalability 81
 - secure mode 24
 - security 26
 - sizing 61
 - sizing tips 60
 - software 29
 - SRA overview 23
 - stickiness 133
 - troubleshooting 155
 - tuning and monitoring 129
 - tuning goals 59
 - typical installation 31
 - usage information 134
 - Portal Server Desktop
 - JavaScript 80
 - NetFile 43
 - portals
 - business intelligence 21
 - collaborative 20
 - overview 19
 - types 20
 - portlet, description 122
 - production environment 129
 - profile database server 86
 - Provider Application Programming Interface 57
 - providers, deployment considerations 126
 - proxies 37
 - and Gateway 86
 - configuration 37
 - failover 84
 - Proxylet, overview 47
 - psdp.dtd file 136
- ## Q
- questions
 - business objectives 49
 - technical goals 51
 - user behaviors and patterns 58

R

- rdmgr command 156
- recovering, Search database 156
- reloading the display profile 157
- requirements, identifying 49
- resource bundles 120
- reverse proxy
 - and application server 151
 - description 119
 - offloading requests 80
- Rewriter
 - load balancing 47
 - overview 46
 - rulesets 46
- Rewriter Proxy
 - and accelerators 75
 - and software crash 84
 - overview 46
- robot 55
- Role-Based Access Control 102
- roles 124
- rulesets, Rewriwter 46

S

- scalability 81
 - and SRA 74
 - portal channels 125
- SDK, description 30
- Search database
 - and robot 55
 - recovering 156
- Search Engine
 - description and benefits 55
 - functions 65
 - structure 96
- search engine
 - sizing factors 64
- search, NetFile 45
- searchURL property 97
- secure mode 24
- securing the operating environment 100
- security 26
 - NetFile 44
 - platform 101
- security strategies 100
- servlets, and communication 85
- session
 - characteristics, SRA 71
 - monitoring 133
 - stickiness 37
- session failover 83, 89
 - and clustering 150
 - BEA 152
- session information 38
- shooter tool 159
- single sign-on 26, 125
 - description 53
 - implementing 125
- sizing 64, 67
 - and JCA 68
 - and JDBC 68
 - establishing baseline figures 62
 - general tips 60
 - Portal Server 61
 - refining 69
 - Search Engine 64
 - search engine factors 64
 - SRA 70
 - tool 71
 - validating 68
- software
 - categories 29
 - packaging 29
 - Portal Server 29
- software crash 84
- Solaris Operating Environment
 - minimizing size of installation 100
 - securing 100
- split tunneling 41
- SRA
 - and load balancing 84, 93
 - and NetFile 44
 - and reverse proxy 119
 - and Sun Enterprise Midframe Line 75
 - components 35, 54

- debugging 158
- directory structure 136
- features and benefits 54
- log files 161
- overview 23, 35
- session characteristics 71
- sizing 70
- troubleshooting 158

SSL

- and Gateway 25
- encryption 100
- Gateway 38
- modes 38
- v2 and v3 38

SSL hardware accelerators 74

state data, and Portal Server services 86

static port applications 40

static portal content 121

static web content 29

subscription channel 55

Sudo 102

Sun Cluster software 88

Sun Crypto Accelerator 1000 board 74

Sun Java System Application Server

- overview 151

SuperAdmin Role 124

system availability 82, 83

system capacity 67

system performance 67

T

- tag library definitions 136
- task list 167
- TCP kernel tuning parameters 146
- technical goals 51
- technical requirements 49
- third party proxy
 - Netlet Proxy 42
- third-party applications
 - and Netlet 41
 - description 122

- thread usage 134
- transaction time 66
- transparent failover, and building modules 94
- troubleshooting 155, ??–162
 - SRA 158
- tuning
 - goals 59
 - Portal Server 129
 - settings 139
- tunneling 41

U

- uniq.pl script 161
- UNIX
 - authentication 43
 - user installation 101
- UNIX processes, troubleshooting 155
- usage information 134
- use case scenarios
 - designing 97
 - example 99
- user behaviors and patterns 58

V

- vertical scaling, description 81
- VPN 54
 - VPN client 41

W

- WAR file 29
 - and application servers 150
 - to deploy software 29
- web containers
 - supported 149
- workload conditions 67

worksheets [163](#)

X

XMLProvider [127](#)

